Computational modeling of observational learning inspired by the cortical underpinnings of human and monkey primates

Emmanouil Hourdakis
March, 2012

University of Crete
Department of Computer Science

Thesis submitted in partial fulfillment of the requirements for the degree of
Doctor of Philosophy

Doctoral Thesis Committee: First member, Professor Panos Trahanias, University of Crete (Advisor)
Second member, Professor Helen Savaki, University of Crete
Third member, Associate Professor Antonis Argyros, University of Crete
Fourth member, Senior Lecturer Yiannis Demiris, Imperial College
Fifth member, Assistant Professor Vasilios Raos, University of Crete
Sixth member, Professor Michael Vrahatis, University of Patras
Seventh member, Professor Michael Zervakis, Technical University of Crete

The work reported in this thesis has been conducted at the Computational Vision and Robotics (CVRL) laboratory of the Institute of Computer Science (ICS) of the Foundation for Research and Technology – Hellas (FORTH), and has been financially supported by a FORTH-ICS scholarship, including funding by the European Commission through project MATHESIS (FP6 IST-027574).

UNIVERSITY OF CRETE

DEPARTMENT OF COMPUTER SCIENCE

Computational modeling of observational learning inspired by the cortical underpinnings of human and monkey primates

Dissertation submitted by

Hourdakis Emmanouil

in partial fulfillment of the requirements for the
Doctor of Philosophy degree in Computer Science

Author: ______________________
Hourdakis Emmanouil

Examination Committee: ______________________
Panos Trahanias, Professor, University of Crete

____
Eleni Savaki, Professor, University of Crete

____
Antonis Argyros, Associate Professor, University of Crete

____
Yiannis Demiris, Senior Lecturer, Imperial College

____
Vasilios Raos, Assistant Professor, University of Crete

Vrahatis
____
Michael Vrahatis, Professor, University of Patras

____
Michael Zervakis, Professor, Technical University of Crete

Departmental approval: ______________________
Panos Trahanias, Professor, Chairman of the Department

Heraklion, March 2012

Acknowledgements

This thesis would never have been completed without the valuable contribution of many people, to whom I'm greatly indebted.

First and foremost I want to express my gratitude to my advisor, Prof. Panos Trahanias, for his guidance, support and patience during the course of this PhD. He has taught me the values of a good researcher and all these years he has been a source of true inspiration and encouragement. I am deeply thankful for the example he has provided as a successful person, researcher and friend.

I would also like to sincerely thank the members of my advisory committee, Assoc. Prof. Antonis Argyros, and Prof. Eleni Savaki, for their support and useful suggestions throughout the course of this thesis. Sincere thanks to Assist. Prof. Yiannis Demiris, Assist. Prof. Vasilios Raos, Prof. Michael Zervakis, and Prof. Michael Vrahatis for participating in the examination committee of this thesis.

My deep appreciation also goes to the Foundation for Research and Technology – Hellas (FORTH) and the University of Crete, where the current thesis was conducted, for providing an excellent and stimulating environment to carry out my research. FORTH has also provided financial support which is gratefully acknowledged.

Many thanks are also due to my colleges at the Computational Vision and Robotics Laboratory (CVRL) for the friendly atmosphere, interesting discussions and all the fun times we had. Special thanks go to Michail Maniadakis, whose support and discussions throughout this PhD were invaluable.

I also want to thank my late father, whose role model of a hard-working and sincere person has given me all the necessary provisions to face life. Even though I miss him, I know that he will always be there in my heart to guide me. A deep thanks to my mother and my two sisters for their support and patience throughout the rough times of this PhD. They have always been there by my side and, even more importantly, they will always be.

Last, but certainly not least, I would like to thank all my good friends and especially my brotherhood friend Mihalis Mpeladakis and my pals Kostis Igoumenakis and Markos Sigalas, who supported me all these years in many different ways. Most importantly, because they were always there for me when I needed them.

Abstract

In the current thesis we have studied the cognitive process of observational learning from a computational modeling perspective. In this context we have employed data from neuroscientific experiments, including higher-level imaging and single-cell recordings, in order to develop two computational models of observational learning, inspired by the neurophysiology of human and Macaque primates. To accomplish this we have devised a framework for designing computational models based on neuroscientific findings, and used it in order to develop two novel implementations of the cortical process in simulated agents. To facilitate learning during observation, both models are based on the intuition that, during action execution and observation, the activated cortical networks in the two primates overlap extensively. As a result, both agents treat perception as an active, cross-modal, simulation of others' actions and learn new motor skills without the active involvement of their body.

The first model maintains adequate consistency with the relevant brain areas and connectivity in Macaques, and effectively provides insights about the cortical underpinnings of observational learning, which can be summarized in three categories: *(i)* neuronal, i.e. how learning can be implemented at the cellular level during observation, *(ii)* regional, by identifying the potential role of a certain region in associating the motor representation with the visual image of the observer, *(iii)* system, how the emergent pattern of activations observed during action observation and action execution is formed, and what are the reasons for the lower activations during observation. In addition, due to the use of the aforementioned modeling methodology, the agent is able to exhibit three important behavioral functions: *(i)* observational learning in a similar manner as its biological counterparts, *(ii)* knowledge generalization to different domains and knowledge integration on top of existing representations and *(iii)* embodiment correspondence based on the overlapping pathway of activations.

The second model employs a phenomenological approach to design a motor control system that is loosely based on the function of the regions that become active in humans during execution and observation. For this reason we have developed novel implementations for each of the subsidiary motor control processes, and integrated them in order to produce an agent able to learn only by observation. The main contributions include: *(i)* a model that replicates the reward prediction properties of the dopaminergic neurons in the Basal Ganglia, used to implement a variant of reinforcement learning, *(ii)* a way to segregate the multidimensional control of the embodiment of the agent to basis functions using a novel primitive model, *(iii)* a method to implement embodiment correspondence using associative networks, which enables an agent to develop and match symbolic representations of its own body and the demonstrator's, *(iv)* how higher-order motor control can be designed as an epiphenomenon of the motor control system, i.e. as a subsidiary process built on top of basis motor functions and *(v)* how learning can be implemented during observation using simple motor rules that can be derived only by observation.

Περίληψη

Στην παρούσα διατριβή μελετήσαμε την γνωσιακή διεργασία της μάθησης μέσω παρατήρησης σε ένα υπολογιστικό πλαίσιο. Για αυτό το λόγο, χρησιμοποιήσαμε δεδομένα από νευροεπιστημονικά πειράματα, όπως για παράδειγμα πειράματα καταγραφών και απόκρισης νευρώνων, ώστε να αναπτύξουμε δύο υπολογιστικά μοντέλα μάθησης μέσω παρατήρησης, εμπνευσμένα από την νευροφυσιολογία των ανθρώπων και πιθήκων Macaque αντίστοιχα. Για να επιτύχουμε τον σκοπό αυτό προτείναμε ένα πλαίσιο για τον σχεδιασμό υπολογιστικών μοντέλων, βασισμένο σε ευρήματα από τις νευροεπιστήμες, το οποίο χρησιμοποιήσαμε ώστε να αναπτύξουμε δύο καινοτόμες υλοποιήσεις της προαναφερθείσας βιολογικής διεργασίας σε προσομοιωμένους πράκτορες. Για να επιτύχουν μάθηση μέσω μόνο παρατήρησης, και τα δύο μοντέλα βασίστηκαν στο ότι κατά την διάρκεια εκτέλεσης και παρατήρησης μιας συμπεριφοράς, τα νευρωνικά δίχτυα που ενεργοποιούνται στον εγκέφαλο των δύο πρωτευόντων ειδών επικαλύπτονται εκτενώς. Ως αποτέλεσμα αυτού του γεγονότος, και οι δύο πράκτορες μεταχειρίζονται την διεργασία αντίληψης σαν μια ενεργή εξομοίωση των συμπεριφορών, και μαθαίνουν νέες συμπεριφορές κίνησης χωρίς να χρησιμοποιούν το σώμα τους.

Το πρώτο μοντέλο διατηρεί ικανοποιητική συνέπεια με τις σχετικές περιοχές και συνδεσιμότητα του εγκεφάλου στους πιθήκους Macaque, και παρέχει πολύ σημαντικές ενδείξεις σχετικά με τα ερείσματα που λαμβάνουν χώρα στον φλοιό των πιθήκων Macaque κατά την διάρκεια της μάθησης μέσω παρατήρησης, που μπορούν να συνοψισθούν σε τρείς κατηγορίες. Σε επίπεδο: *(ι)* νευρώνα, εξηγώντας πώς η μάθηση μπορεί να υλοποιηθεί σε κυτταρικό επίπεδο κατά την διάρκεια παρατήρησης, *(ιι)* εγκεφαλικής περιοχής, με το να αναγνωρίσει τον δυνητικό ρόλο μιας συγκεκριμένης περιοχής στο να συσχετίζει τις κινητικές αναπαραστάσεις με την οπτική εικόνα του παρατηρητή και *(ιιι)* συστήματος, δηλαδή πως σχηματίζεται το αναδυόμενο σχέδιο ενεργοποιήσεων που έχει παρατηρηθεί κατά την διάρκεια εκτέλεσης και παρατήρησης μιας συμπεριφοράς, και τους λόγους της χαμηλότερης ενεργοποίησης κάποιων περιοχών κατά την διάρκεια παρατήρησης. Επίσης, λόγω της χρήσης της προαναφερθείσας μεθοδολογίας, ο πράκτορας μπορεί να επιδείξει τρεις σημαντικές συμπεριφορές: *(ι)* μάθηση μέσω παρατήρησης, με τρόπο παρόμοιο με αυτόν που χρησιμοποιούν οι αντίστοιχοι βιολογικοί ομόλογοι του, *(ιι)* γενίκευση γνώσης σε διαφορετικούς τομείς και ολοκλήρωση γνώσης χρησιμοποιώντας προϋπάρχουσες αναπαραστάσεις και *(ιιι)* αντιστοιχία του σώματος του με άλλα σώματα χρησιμοποιώντας ως βάση το μονοπάτι των επικαλυπτόμενων ενεργοποιήσεων.

Το δεύτερο μοντέλο χρησιμοποιεί μια φαινομενολογική προσέγγιση για να σχεδιάσει ένα κινητικό σύστημα που βασίζεται χαλαρά στην λειτουργία των περιοχών που ενεργοποιούνται στους ανθρώπους, κατά την διάρκεια εκτέλεσης και παρατήρησης μιας συμπεριφοράς. Για αυτό το λόγο, αναπτύξαμε μια καινοτόμα υλοποίηση για κάθε μια από τις δευτερεύουσες κινητικές διεργασίες, και τις ενσωματώσαμε ώστε να παραχθεί ένας πράκτορας που μπορεί να μάθει μέσω μόνο παρατήρησης. Ανάμεσα στις βασικές συνεισφορές της δουλειάς περιλαμβάνονται: *(ι)* ένα μοντέλο που αναπαράγει τις ιδιότητες πρόβλεψης της ανταμοιβής που έχουν οι ντοπαμινεργικοί νευρώνες στα βασικά γάγγλια, *(ιι)* ένα τρόπο για να διαχωρίσουμε το πολυδιάστατο έλεγχο του σώματος του πράκτορα, *(ιιι)* μια μέθοδο για να αντιμετωπίσουμε το πρόβλημα αντιστοιχίας του σώματος χρησιμοποιώντας νευρωνικά δίχτυα συσχετισμού, *(iv)* πώς ο υψηλότερης τάξης κινητικός έλεγχος μπορεί να σχεδιαστεί επιφαινομενικά, βάσει του κινητικού συστήματος, ως επικουρική διεργασία χτισμένη από πάνω από άλλες βασικές κινητικές λειτουργίες και *(ν)* πως η μάθηση μπορεί να υλοποιηθεί κατά την διάρκεια παρατήρησης χρησιμοποιώντας απλούς κινητικούς κανόνες που μπορούν να εκμαιευτούν μόνο από παρατήρηση.

Contents

List of figures

List of Tables

Chapter 1

Introduction

Observational learning is the ability to learn and acquire new motor knowledge only by observation. Otherwise referred to as mental practice, or mental imagery, observational learning is formally defined as *the symbolic rehearsal of a physical activity in the absence of any gross muscular movements* (Richardson, 1967). In primates, mental imagery is a very important skill, and a core component of the cortical network that allows them to understand and reproduce meaningful cross-modal communication (Zagacki et al., 1992). Recently, neuroscientific findings have shed more light on the cortical underpinnings of this process, by discovering that the network of brain regions that is used for action execution overlaps extensively with the one used for action observation (Raos et al., 2004; Raos et al., 2007; Evangeliou et al., 2008; Kilintari et al., 2010). Inspired from this finding, in the current thesis, we have developed computational agents that can learn during observation, i.e. without the active involvement of their body.

In the context of computational modeling, the implementation of observational learning is an important step towards the development of robots that can reproduce meaningful behavior by social interaction. Traditionally, in the computer science literature, social learning has been studied based on imitation, i.e. by physically interacting with others (Schaal et al., 2003; Wolpert et al., 2003; Ijspeert et al., 2002a). In contrast, mental imagery can enable an ongoing, inter-subjective, learning process during which an agent continually acquires new knowledge from its environment without the need to interact with it physically.

In primates, capacities for imitation have been reported for as early as the third year of an infant's life (Meltzoff and Moore, 1977). Evidently, while still during infancy, humans have already developmentally acquired a formal representational system of their external world, capable of integrating information from all senses and processing percepts (Fig. 1.1). The fact that this ability is found so early in an infant's life is the result of both ontogenetic and epigenetic development (Jones, 2009), and pinpoints the importance of observation for the socialization of the species.

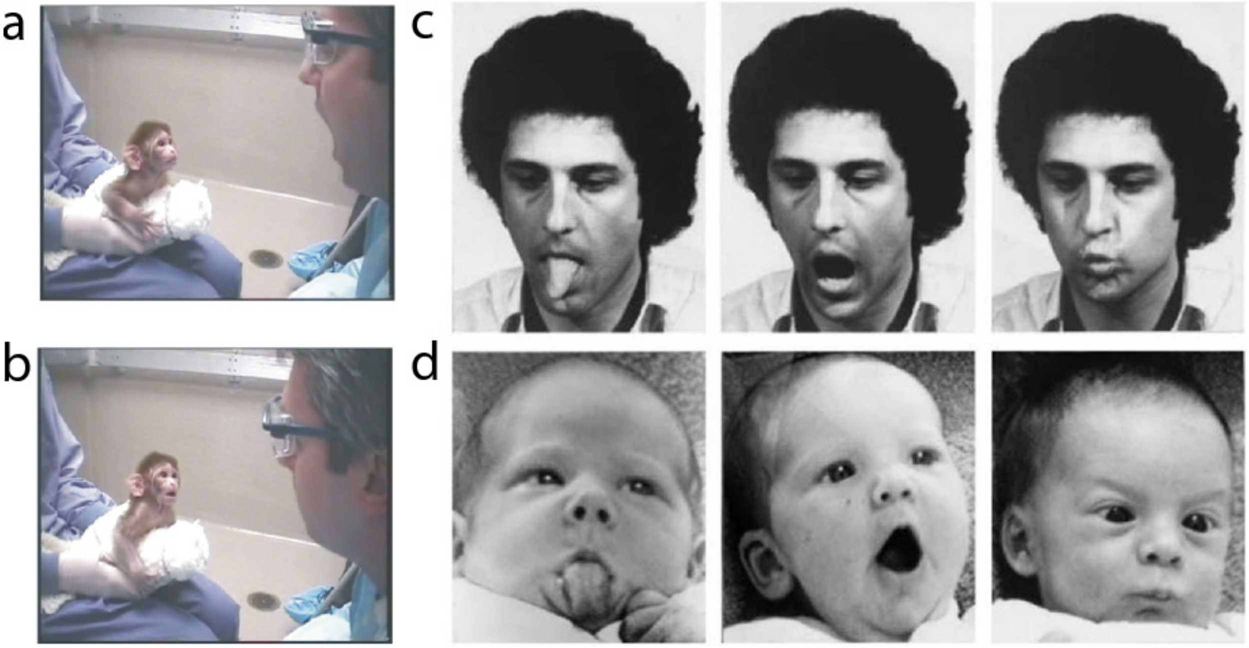


Fig. 1.1. Two-year old infants can exhibit imitation skils. a,b: Macaque infants imitate lip smacking movements. c.d.: Two to three week old human infants imitating facial gestures. Images adopted from (a, b: Ferrari et al., 2006) and (c, d: Meltzoff and Moore, 1977).

Similarly, in computational modeling, observational learning can provide a framework for developing artificial agents that can acquire new skills only by observing the actions of their counterparts, thus broadening the possible ways that robots can socialize. More importantly, modeling the underpinnings of this system will open new ways for robotics to implement inter-subjective communication and different forms of cross-modal interaction, leading to novel methods for associating perception and cognition, integrating different senses together, and assigning meaning to a behavior.

1.1 The cognitive explanation of the cortical underpinnings of overlapping pathways

Mental imagery is a high-order cognitive function, i.e. it is realized through a number of conscious and subconscious processes in the brain. Insights for its cortical underpinnings come from recent neuropsychological studies, which have investigated the neural activity in the cerebral cortex of human and monkey primates during action observation and action execution (Raos et al., 2004; Raos et al., 2007; Kilintari et al., 2010; Evangeliou et al., 2008). Results have identified a common representational substrate between motor observation and motor execution, suggesting that there is a close functional equivalence between the two processes (Jeannerod and Johnson-Frey, 2003). In monkeys, high resolution imaging experiments were able to identify a network of regions that actively participate in execution and observation, extending from visual (Kilintari et al., 2010) to somatosensory and motor regions (Raos et al., 2007; Raos et al., 2004; Kilintari et al., 2010; Evangeliou et al., 2008; Savaki 2010). Similar findings have been reported for human primates, where researchers have identified an analogous network of active regions, involved in motor preparation and planning (Caspers et al., 2010).

The aforementioned neuroscientific evidence suggests that in both monkey and human primates, perception of motor actions is mediated by motor knowledge (Viviani and Stucchi, 1989). The cognitive interpretation of this mechanism is that, when we observe an action performed by a conspecific, we activate our own motor system in order to understand it. Consequently, our grounded motor experiences, i.e. the rules and knowledge that we have accumulated throughout our interactions with others and the environment, are used as a substrate in order to understand and perceive an observed action (Decety and Ingvar, 1990). This striking property of our perceptual system is also responsible for our ability to learn during observation. Motor mental images include a representation of the body of the demonstrator, rather than just the consequences of its actions (Jeannerod, 1994), indicating that perception is embodied and modal. For this reason, motor imagery has been established as an important method for improving motor performance (Driskell et al., 1994), while several studies have associated it with active skill learning (Finke, 1980; Denis, 1985) and muscle strength increase (Yue and Cole, 1992).

Since observation and execution activate the same conceptual representations, to understand the content of mental motor images one must look more closely into the functions of the motor control system. The overlapping pathways discovered by the aforementioned studies (Raos et al., 2007; Raos et al., 2004; Kilintari et al., 2010; Evangeliou et al., 2008; Savaki 2010) suggest that motor execution and motor imagery share a common representational substrate (Jeannerod, 1994). In the former case action execution is performed covertly, using the body of the agent, while in the latter, the motor act is rehearsed mentally, without producing any overt motor output. Similarly to visual imagery, which pertains to the representations of a visually perceived object, motor imagery pertains to the motor physiology of the agent (Jeannerod and Johnson-Frey, 2003; Decety et al., 1997). This convention provides an important indication on how motor control is structured in the primate brain, and in the current thesis it is explored in order to develop computational agents that support observational learning.

1.2 How observational learning can help towards developing social robots

Our ability to adapt to our social environment is one of the primary components behind the evolution of our intelligence (Barresi and Moore, 1995). For primates, socialization is an inter-subjective process during which an agent learns to develop concepts from its environment based on the interaction with others (Box and Gibson, 1999). Consequently, the cognitive processes that underpin perception and action understanding, i.e. our ability to perceive and assign meaning to the behaviors of others, constitute a major component for developing social cognition (Blakemore et al., 2004).

In the literature, social psychologists and anthropologists have proposed two theories in order to explain the underlying principles behind socialization, the Theory of Mind (ToM, Gordon 1986; Heal 1986) and the Theory-Theory (Carruthers and Smith 1996). Their formulation has given impetus for an extended debate regarding socialization and how it is facilitated in primate cognition, mostly because both suggestions have a strong theoretical basis but lack evaluation from experimental evidence. Theory-theory suggests that our mind possesses the core of a 'folk' psychological theory, which is epigenetically encoded and enables us to understand social events such as perceiving, assigning meaning to actions and identifying intentions. In contrast, ToM takes a simulation stance, and suggests that to understand others we actively simulate their actions using our own experiences (Gallese and Goldman, 1998). By imagining ourselves in the place of the social partner we are interacting with, we can detect its intentions and anticipate its future actions. Therefore, humans do not only perceive, sense and move, but can also activate their own conceptual representations for perceiving, sensing and moving when observing others. The core of this ability lies in the capacity of primates to perform mind reading (Baron-Cohen, 1997; Leslie and Thaiss, 1992), i.e. use their own grounded experiences to make behavioral predictions about others' actions (Perner, 1991).

Behaviorally, ToM is interlinked with the functional system that supports executive control (Perner and Lang, 1999). For example children with autism spectrum disorder are unable to exhibit ToM because they suffer from basic motor dysfunctions (Ozonoff et al, 1991). Using the same line of thought, Russel (Russel, 1996) has argued that executive function is important for the development of self-awareness, and therefore a prerequisite for ToM. Even though capacities that could facilitate the ToM have been observed in young infants (Wimmer and Perner, 1983), its cortical underpinnings are largely unknown. Until recently, the major evidence came from the existence of mirror neurons, a class of visuo-motor cells, in the Macaque's ventral premotor cortex, that facilitate action recognition (Rizzolatti et al., 1996). The findings discussed in section 1.1 above, not only provide cortical evidence for the existence of ToM in primates (Raos et al., 2007; Raos et al., 2004; Kilintari et al., 2010; Evangeliou et al., 2008; Savaki 2010), but go one step further to unravel its basic mechanisms: understanding others involves a complete reenactment of their sensorimotor representations (Savaki, 2010).

In addition, the evidence for overlapping pathways has another important implication: it can refine basic conceptions about cognition. The active involvement of the primates' somatosensory areas during observation (Raos et al., 2004) provides evidence in favor of contemporary theories that view cognition as embodied (Dautenhahn, 1996; Anderson, 2003). It suggests that we understand others by simulating the consequences of what we observe up to the level of our sensory systems. This makes an important statement about the relationship between our mind and our body, proposing that they co-develop and are closely interlinked. In addition, the existence of this shared neural substrate provides evidence on how the stimuli from different senses are integrated within our cognitive system. Vision, hearing and touch are not passive perceptual events, but active inter-modal simulations of our own somatosensation codes. In other words, even though perception is modal, it is encoded in multi-modal conceptual representations.

It is therefore evident that modeling the process of observational learning based on the cognitive mechanisms described above will open new ways to implement social skills in robots, i.e. develop agents that can assign conceptual meaning to human behavior. Moreover, the cognitive hypothesis of a shared representational system for perception and action can extend to other domains and provide an important functional basis for implementing social processes, such as empathy, listening, response facilitation and agency attribution.

1.3 Scope of the thesis

From the discussion in the previous sections it is evident that observational learning can give a great impetus towards developing robots that possess the ability to interact and learn by observing others. In the current thesis, we have examined how computational agents can be designed to support this ability using inspiration from the cortical and neural underpinnings of the process in the brain of primates. To accomplish this, we have focused on two species that possess observational learning capacities: humans and monkeys. To understand how this function is performed cortically, we have examined the large body of neuroscientific studies that pertain to observation and execution, and derived novel computational implementations inspired by the two neurophysiological models (Hourdakis et al., 2011; Hourdakis and Trahanias, 2011a; Hourdakis and Trahanias 2011b; Hourdakis and Trahanias, 2011c; Hourdakis and Trahanias 2011d; Hourdakis and Trahanias, in press b). In both cases, our working hypothesis was the fact that, in the two species, a common neural pathway between execution and observation is being activated (Raos et al., 2007).

One important distinction between the neurophysiological models of the two primates is in the resolution of the available data. In monkeys, due to the fact that single cell penetrations are permitted, neuroscientific data can be gathered up to the level of single neurons. In contrast, in human subjects, available data are extracted from imaging studies and therefore are limited to providing a spatial resolution up to the level of neuronal populations. To compensate for this fact, we have developed two models. The first one, which was based on Macaque monkeys, was designed to be consistent with certain cortical processes that take place in the brain of the species during observational learning and can be described up to the level of single cells (Hourdakis et al., 2011; Hourdakis and Trahanias, 2011b). To guide model development we combined data from all available neuroscientific sources, including lesion studies, higher-level imaging data and single-cell recordings, in order to derive a computational model that is consistent with the neuronal properties that have been identified as important to the process of observational learning. In the case of the second, human model, regional activation data provided us with general information regarding the areas that participate in observational learning, and the extent to which each area is correlated with a specific cognitive function. To compensate for the limited amount of available information, the second model was designed using a phenomenological approach (Arkin, 1998), i.e. by focusing on the behavioral aspects of the cortical process, rather than its exact biological details. For this reason, we've identified the abstract roles that are carried out by the overlapping regions during execution and observation, and combined them in order to facilitate observational learning.

1.4 Theoretical questions that rise from the computational implementation of observational learning

Our working assumption for the development of the two models is that there is a close functional equivalence between motor observation and motor execution. The research discussed in the previous section, as well as a plethora of studies which we will review in later chapters, provide evidence on the cortical underpinnings of this phenomenon by suggesting that the two processes use the same neural systems. This finding provides an important framework for examining how motor observation is facilitated in both human and monkeys: *The fact that mental and overt motor images share the same neural structures suggests that action observation employs, to a certain extent, the cognitive functions of motor control* (Savaki, 2010). This includes a large category of cortical processes that pertain to body perception, goal representation, higher-order control of movement and motor execution. In order to develop computational agents that can facilitate observational learning it is important to understand the contribution of all these processes during observation. Consequently, the development of the two models was guided by three theoretical questions, which pertain to how the process of observational learning can be modeled within a computational context. These are outlined and briefly discussed below.

A. *What are the representations in a mental imagined state and how can they be coupled with the motor control system?*

Undoubtly, the most important issue that one must look into prior to developing a computational model of a cortical process, is the type of representations that will be used to describe it. In our case, this pertains to the representations of the mental imagined state, i.e. what the agent will perceive when observing others' actions. The evidence that we discuss in this thesis suggests that, in primates, this representation is obtained by converting the visual image from the observation of the demonstrator to an overt mental image, using the agent's motor control system (Raos et al., 2007; Raos et al., 2004; Kilintari et al., 2010; Evangeliou et al., 2008; Savaki 2010). Computationally, the involvement of the motor control system means that various important problems pertaining to embodiment correspondence (Nehaniv and Dautenhahn, 2002), sensorimotor integration (Tin and Poon, 2005) and state representation (Kawato, 1999) must also be considered. Moreover, a computational model of observational learning must be able to suggest a way for these processes to become active and contribute to the formation of the mental image, without the active involvement of the agent's body.

To confront this problem, we focused on the fact that action observation and action execution share the same neural representation codes (Savaki, 2010). Consequently, in both models, observation was refined as an active perceptual process, during which the observer was able to associate perceived actions, obtained from vision, with the representations of its own cognitive system. Moreover, the fact that the motor component is activated during observation suggests that the mental image of an observed movement is tightly coupled to the motor image of the observer (Jeannerod, 1994). To understand this association we have derived novel implementations for the motor system of the two agents, based on contemporary theories of motor control. This involved the development of models for state representation, state estimation, motor control and, in the case of the human model, higher-order control and reward processing. Moreover, we have suggested how these subliminary processes can be actively employed during observation of a movement, using the internal simulation of the motor control system.

B. *How can learning be facilitated without the active involvement of the body of an agent?*

The second question regards to *how learning can be facilitated during observation alone*, i.e. without the active involvement of the agent's body. This is a challenging issue in robotic research, since all contemporary approaches for learning are based on knowledge grounding (Harnad, 1990), i.e. learning by direct interaction with the environment (Ijspeert et al., 2002a; Billard, 2000). To implement learning only by observation, one requires an understanding of the cortical underpinnings of the neurophysiological model at various resolution levels: *(i)* at the neural level, one must resolve how learning can be implemented in the synapses between neurons, *(ii)* at the cortical level, how different regions can facilitate plasticity, while *(iii)* at the behavioral level how can motor control be structured accordingly so that an agent can learn from simple percepts derived only by observation.

All three issues are investigated thoroughly in the two developed models. In the first case, the high modeling resolution of the Macaque imaging data enabled us to understand how plasticity can be modeled at the synaptic and regional level of our computational agent, given the overlapping neural pathways during observation and execution. More specifically, we have used biologically inspired neuron and synapse models to develop the computational agent, and found that learning can be facilitated due to a synchronization in the neural responses of certain regions. At the behavioral level, the second model, using a phenomenological approach, implements a hierarchy of motor control functions, which enable it to learn in the peripheral components of its motor system, using simple control rules that can be derived only by observation.

C. *What is the neurophysiological basis of the mechanism that is responsible for activating the sensorimotor representations in primates during observation?*

The activity reported in primates during action observation is caused by the top-down modulation of the cortical regions that respond to the observer watching an action. To understand the neural underpinnings of these overlapping pathways one important issue that must be resolved pertains to *the neurophysiological basis of the mechanism that activates the sensory and motor control centers of the agents during observation*. Put more simply, what are the reasons that cause the motor control system of an agent to become active when observing an action? To answer this question one must develop an understanding of the top-down and bottom-up pathways that are responsible for causing these activations, both cortically and computationally.

The high resolution of the Macaque model enabled us to evaluate important assumptions regarding the reasons for the overlapping activations. The agent that was developed exhibited a tendency to activate the same motor areas during execution and observation emergently, a phenomenon that was attributed to the connectivity between the computational regions. More importantly, the agent using models of biologically inspired neurons and synapses, was able to replicate, to a certain extent, the levels of activation during observation found by neurophysiological studies in the Macaque brain. The second model, which was designed in a lower resolution, used a behavioral approach to suggest that the activation of the overlapping pathways during observation and execution could be caused as an epiphenomenon of the activation of the motor control system during observation.

The three theoretical questions were used as a substrate in order to guide model development, and are explicitly addressed during the description of both models. In the following section we outline the major contributions of the work, as well as the list of Journal and Conference publications that were produced.

1.5 Thesis Contributions

In the current thesis we have studied, developed and experimentally verified two computational models of observational learning, inspired by the neurophysiology of human and Macaque primates. To accomplish this, we have devised a framework for designing computational models based on neuroscientific findings, and used it in order to develop two novel implementations of the cortical process in simulated agents. To facilitate learning during observation, both models were based on the intuition that, during action execution and observation, the activated cortical networks overlap extensively. This constitutes a novel way for designing computational models of motor learning, by treating perception as an active, cross-modal, simulation of others' actions. As a result of this, the two developed computational agents can learn new motor skills without the active involvement of their body. To the best of our knowledge, this is the first attempt to model this process computationally, as all contemporary approaches to motor learning focus on imitation, which involves the use of an agent's body.

Further to the above, contributions of this thesis include a novel methodology for designing and implementing biologically inspired computational models, a method to quantify the quality of a biologically inspired neural network, as well as two computational models of observational learning inspired by neurophysiological data.

The first model provided very important indications about the cortical underpinnings of observational learning in Macaque primates, which can be summarized in three categories:

(i) Neuronal, i.e. how learning can be implemented at the cellular level during observation.

(ii) Regional, by identifying the potential role of a certain region in associating the motor representation with the visual image of the observer.

(iii) System, how the emergent pattern of activations observed during action observation and action execution is formed, and what are the reasons for the lower activations during observation.

In addition, due to the use of the aforementioned modeling methodology, the agent was able to exhibit three important behavioral functions:

(i) Observational learning in a similar manner as its biological counterparts.
(ii) Knowledge generalization to different domains and knowledge integration on top of existing representations.
(iii) Embodiment correspondence based on the overlapping pathway of activations.

The second model employed a phenomenological approach to design a motor control system that is loosely based on the function of the regions that become active in humans during execution and observation. For this reason we have developed novel implementations for each of the subsidiary motor control processes, and integrated them in order to produce an agent able to learn only by observation. The main contributions are outlined below:

(i) A model that replicates the reward prediction properties of the dopaminergic neurons in the Basal Ganglia, used to implement a variant of reinforcement learning.
(ii) A way to segregate the multidimensional control of the embodiment of the agent to basis functions using a novel primitive model.
(iii) A method to implement embodiment correspondence using associative networks, which enables an agent to develop and match symbolic representations of its own body and the demonstrator's.
(iv) How higher-order motor control can be designed as an epiphenomenon of the motor control system, i.e. as a subsidiary process built on top of basis motor functions.
(v) How learning can be implemented during observation using simple motor rules that can be derived only by observation.

1.5.1 List of publications

The aforementioned contributions and relevant results have been published in scientific journals and presented in peer reviewed conferences as follows:

Journal Publications:

1. Invited Journal Submission: E. Hourdakis and P. Trahanias, Improving the performance of Liquid State Machines based on the separation property, *under review by the Journal of Neurocomputing*.
2. E. Hourdakis and P. Trahanias, "Observational learning inspired by human primates", *under review by the Journal of Adaptive Behavior.*

3. E. Hourdakis, E. Savaki and P. Trahanias, “Computational modeling of cortical pathways involved in action execution and action observation”, Neurocomputing, vol. 74 , Issue 7, pp.1135-1155, 2011.

Conference Publications:

4. E. Hourdakis and P. Trahanias, ”Improving the performance of liquid state machines based on the separation property”, *Engineering Applications of Neural Networks, EANN11, Corfu, Greece, 2011*. **Awarded with the best student paper award**.
5. E. Hourdakis and P. Trahanias, “Observational learning based on models of overlapping pathways”, *International Conference on Artificial Neural Networks, ICANN11, Helsinky, Finland, 2011*
6. E. Hourdakis and P. Trahanias, “Observational learning based on overlapping pathways”, *Second International Conference on Morphological Computation, MORPHCOMP11, Venice, Italy.*
7. E. Hourdakis and P. Trahanias, “Computational modeling of online reaching”, *European Conference on Artificial Life, ECAL11, Paris, France*
8. E. Hourdakis and P. Trahanias, “A framework for automating the construction of computational models, *Congress on evolutionary computation*, *Norway* 2008.
9. E. Hourdakis, M. Maniadakis and P. Trahanias, “A biologically inspired approach for the control of the hand”, *Congress on Evolutionary Computation, Singapore 2007.*
10. M. Maniadakis, E. Hourdakis and P. Trahanias, “Modeling overlapping action execution/observation brain pathways”, *International Joint Conference on Artificial Neural Networks*, *IJCNN, Atlanta 2007.*

1.6 Thesis outline

To facilitate readability and comprehensibility, the thesis is structured in 7 chapters. In the current section we briefly outline the content of each chapter, by providing a comprehensive overview of all the topics that will be discussed.

In chapter two, we visit the large body of literature that is related to observational learning and motor control. We introduce this review by discussing about the established methods in biologically inspired brain modeling, in order to give the reader a comprehensive overview of the contemporary approaches in the field. In this context, the discussion is mainly focused towards:

i. Popular methods used to image the brain and how these are mapped in different model resolutions.
ii. The extent to which a computational implementation of a biological process can be considered as a valid model of the process itself.

Having established a solid theoretical framework for computational modeling, we shift our discussion towards biologically inspired methods for modeling motor control. Due to the interdisciplinary nature of the processes that are involved in designing a motor control system, the review covers a wide range of topics such as:

i. Primitives and the importance of modularity in motor control.
ii. Internal models and issues regarding state representation and state estimation.
iii. Policy learning and planning in motor control systems.

All these processes are vital for the development of a motor control system. To implement learning in this system, one must look into additional topics that pertain to embodiment correspondence and motor learning by imitation, which are also reviewed in this chapter. Finally, we conclude the discussion by visiting the most popular biologically inspired motor control models in the literature, and analyze their strengths and weaknesses.

Chapter three provides a complete overview of the cortical underpinnings of observational learning in human and monkey primates. We first look into phenomenological reasons that can explain the overlapping pathway of activations during action execution and action observation. We then continue to investigate the cortical functions that are performed in the regions that become active during observation, by examining interdisciplinary data from various neurophysiological studies. The chapter concludes by analyzing the social functions that have been attributed to the overlapping pathways.

Chapter four introduces the computational modeling approach that was used to develop our agents. For this reason it briefly discusses topics such as the brain structure, neural codes, functional areas, modularity and cortical representations. Having established a comprehensive overview of the subject, we then discuss our own modeling approach, based on the concept of *computational pathways*, and how it was employed in order to develop the two artificial agents. Finally, we describe the biologically inspired neural network that was used to develop the second agent, and suggest a measure that can quantify its quality by evaluating the network’s ability to separate different datasets.

Chapter five presents the first of the two computational models, which was inspired by neurophysiological data from studies in Macaque monkeys. It starts by discussing the cognitive abilities that enable Macaques to learn by observation, and how the overlapping pathway of activations can facilitate this function. In addition, the chapter describes the modeling approach that was used to implement the neurophysiological model and provides details for the synapse and attractor neural networks that were used. It continues to discuss the intuition behind the development approach, by looking into how a neural code can be represented computationally, as well as various methods from the literature that pertain to its implementation.

Having presented all the underlying tools that were used to develop the model, the chapter continues to discuss the development of a computational agent that is based on the neurophysiology of the Macaque monkey during observation. In this context, we describe how the different motor components of the agent are implemented computationally and integrated together in order to produce a working model of observational learning. Finally, we carry out an extensive evaluation of the agent's ability to learn only by observation, in a manner that is similar to its biological counterparts.

Chapter six presents the second model, which was based on the neurophysiology of human primates. For this reason, it discusses the cognitive functions and the types of learning that humans can afford, paying special attention to their ability to learn new skills only by observation. We then continue to describe the role of the areas that have been found by neurophysiological studies to be active during observation, and suggest a novel method to structure the motor control system in order to enable learning during observation. We continue to discuss the development of the computational agent, focusing on how the proposed theory of pathways can allow us to integrate the different functions that are carried out during motor control. In this context, we present the four submodels that were designed in order to enable the agent to perform motor control. Having described the development of the model we then provide an extensive evaluation of the computational agent's ability to learn new skills only by observation.

The final chapter of the thesis revisits the two models presented, and discusses the various issues that were raised during their implementation. More specifically we focus on the modeling resolution of the two models and the types of assumptions that can be validated by each one. In addition we place the two models in the literature by drawing comparisons, where applicable, with other works, and discuss how the current work has progressed the state of the art. The section also provides an evaluation of the pathway modeling approach that was used to implement the two models, and how it can be employed as an extended method for modeling biologically inspired data from neurophysiological studies. Moreover, we discuss the importance of observational learning for *(i)* developing robots that can fully interact with their environments, and *(ii)* facilitating the emergence of social cognition in computational agents. The chapter is concluded by outlining directions for future work, and in particular how the two models can be extended in order to: *(i)* consider additional processes that are associated with observational learning such as agency attribution and motor inhibition, *(ii)* provide a basis for the development of social processes in robotics, including concepts that pertain to intersubjective communication and empathy.

Chapter 2

Literature Review

As briefly outlined in the introduction, observational learning is a higher-order cognitive ability, which is extensively based on motor control and involves a large number of auxiliary cortical functions. Each of these functions contributes in a different way to the process, by carrying out tasks such as state representation (Kawato, 1999), embodiment correspondence (Nehaniv and Dautenhahn, 1999), symbol grounding (Harnad, 1990) and motor control (Billard, 2000). In the robotic literature, since the early nineties, many researchers have realized that biology can provide working conceptual models for each of these problems, and focused on designing computational agents that can solve them based on principles from neuroscience. This trend led to a number of models (Schaal, 1999; Schaal et al., 2003; Dautenhahn and Nehaniv, 2002) at different resolutions levels (Webb, 1991), which employed biological principles in order to suggest new methods for solving difficult engineering problems.

In the current section we provide a comprehensive overview of the literature of computational modeling of motor control processes. We first discuss issues regarding model resolution and level of detail, in order to give the reader a perspective on the extent to which a computer simulation can be considered as a replica of a biological process. We then continue to review the most recent works in computational modeling of motor control, since it is closely related to observational learning. For this reason we discuss state of the art models, which solve problems that are associated with motor control, such as embodiment correspondence, primitive functions, internal models, policy learning and higher-order motor control. Even though we do not entirely exclude other approaches, our main focus is on models implemented via neural networks, because they follow principles that are similar to the brain's organization (Amit, 1989).

2.1 Computational modeling of natural processes

An increasing number of researchers nowadays is focusing on extracting basic principles from biological systems in order to suggest new solutions for engineering problems (Webb, 2000). This trend is supported by recent advances in imaging neuroscience, which have enabled scientists to understand many of the properties underlying the function of the nervous system.

In the context of engineering, a biologically inspired model is a model that replicates, in simulation, the underpinnings of a natural process (Fig. 2.1). The benefits from modeling natural processes are twofold (Webb, 1991). On one end, computer scientists can employ principles from natural systems in order to confront difficult engineering problems. To accomplish this, computational models focus on developing working solutions of a given problem, rather than replicating the precise physiological structure of the underlying biological process (Zipser, 1992). On the other end, neuroscientists are offered the capability to employ robots, due to their analogy with animal physiology, in order to gain more insights into the cortical underpinnings of a biological mechanism (Dean, 1991). In this case, the modeling resolution that is used is high, in an attempt to capture the details of the cortical process under investigation.

In the literature, these two opposing views have generated a long standing debate regarding the correct level of correspondence between a model of a biological process and the process itself (Webb, 2000). Some argue that a simulation of a process is bound to be misleading, since it cannot capture the real world complexity (Mataric and Cliff, 1997). Others criticize over-complexity, deeming it as unable to produce robust, working solutions to a problem (Koch, 1999; Maynard, 1974).

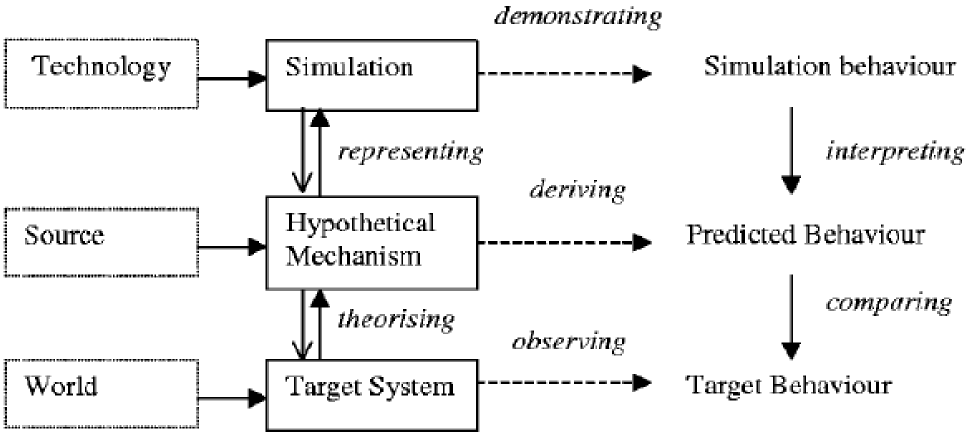


Fig. 2.1. The different stages of development of a biologically inspired model. The simulation is based on the properties of the source process, and must produce results that are verifiable by the target behavior. Image adopted from (Webb, 2000).

In our opinion, it is neither the focus on detail or abstraction that ensures the quality of a model. Indeed, a model faithful in every detail to a biological process is bound to be incomprehensible as the biological process itself, and will not contribute anything new to the understanding of the system (Collin and

Woodburn, 1998). Webb (Webb, 2000) has argued that there are five dimensions on which a simulation model can be judged:

- *(i)* Relevance, i.e. whether it is able to generate hypotheses that are applicable to biology.
- *(ii)* Level, defined by its elementary units.
- *(iii)* Generality, i.e. the range of biological systems that it can represent.
- *(iv)* Structural accuracy, or how well it represents the mechanisms underlying the behavior.
- *(v)* Performance, i.e. to what extent the model behavior matches the target behavior.

In practice, there isn't any specific point in the space of these parameters that defines a good model, since there are fundamental tradeoffs when benefiting one quality over another (Levins, 1966). Instead, a good model is well defined if it can explicitly identify its position in each of the five aforementioned dimensions. All computational models must be abstracted from the underlying biological process they are replicating so that they can be computationally tractable. It is however important that this abstraction is clearly expressed, and reflected in the extent to which they make predictions about the modeled process.

In the next section we review approaches in imaging neuroscience, in order to give the reader a perspective on the types of data, and the methods used to obtain them, that are available to computational modelers.

2.1.1 Functional NeuroImaging

Until recently the only way to investigate the underpinnings of primate cognition was either through brain lesioned patients (Geschwind, 1965), or by studying the electrical signals of patients going through neurosurgery (Penfield and Rasmussen, 1938). This picture however has changed in recent years, since new methods in functional neuroimaging have produced a vast amount of data regarding the function of the brain (Fristen, 1997; Roland, 1993; Posner and Raichle, 1994).

To decipher the brain function it is important to gain a deep understanding of its workings at many different levels, from cellular to regional and interregional. An impetus towards this goal is given by the wealth of data that has been made available by neuroscientists. This is evident from several conferences (ICCI, BVAI, CNS, ICCM, ALIFE X), scientific journals (Neurocomputing, Neural Computation, Cognitive Neuroscience, Brain Research, Behavioral and Brain Sciences) and workshops (CompMod, CSMS, Brain Connectivity), that focus on modeling results from such imaging studies, in order to build systems that are inspired from the functioning of all kinds of biological species.

The first breakthrough towards understanding the brain came at the beginning of the century, when neuroscientists incorporated staining methods such as cytoarchitectonics in order to segment the cerebral cortex into topographic maps (Brodman, 1905), based on the regional diversity of its histological structure (Fig. 2.2). More recently, functional neuroimaging offered new insights to the way these spatially arranged regions contribute to different brain processes. Using techniques such as PET

(Ter-Pogossian and Phelps et al. 1975), fMRI (Ogawa and Lee, 1990), MRI (Mattson and Simon 1996), MEG (Cohen, 1968), ERP (Rohrbaugh, Parasuramanan et al. 1993) and ^{14}C-Deoxyglucose (Sokoloff, Reivich et al. 1977), among others, scientists were able to unveil aspects of brain processing by investigating the activation history of a cortical region and determining its contribution to a cognitive process.

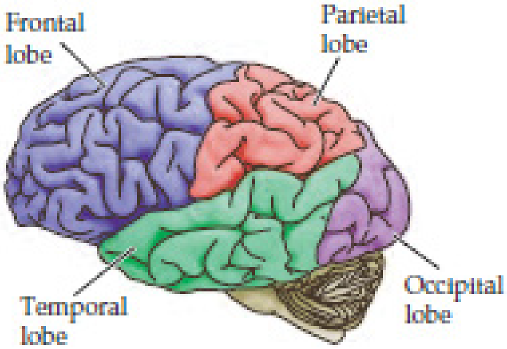


Fig. 2.2. The four major subdivisions identified in the human brain, *(i)* frontal, *(ii)* parietal, *(iii)* temporal and *(iv)* occipital. Each lobe, marked with a different color, is specialized in a different form of processing. Image adopted from (Purves et al., 2001).

Imaging studies can be classified into two types based on the nature of signals that they monitor in the brain. On one end, there are methods that measure the electric basis of a neural activity (magentoencephalography and event related potentials) while on the other, those that measure the metabolic increase in the brain (positron emission tomography, functional magnetic resonance imaging and ^{14}C-deoxyglucose). Each imaging type differs in its spatial and temporal resolution. For example fMRI signals can provide a spatial resolution of 2mm and temporal resolution in the order of seconds, while PET is restricted to 5mm spatial resolution and temporal in tens of seconds. Practically, this means that certain events in the brain may only be observable by a method that has the appropriate resolution. For example in Macaques, ^{14}C-deoxyglucose experiments (Raos et al., 2007; Raos et al., 2004) were able to identify an extended overlap between regions during action observation and action execution, due to the fact that they have a resolution of two orders of magnitude higher (20 microns).

Nonetheless, all the aforementioned methods can reach at best a spatial resolution of 5mm and a temporal resolution of a few seconds. This means that events in the brain that are in the order of milliseconds, such as the firing activity of individual cells, are not directly observable by any of the above methods. To compensate for this, researchers carry out single cell recording experiments, where they penetrate the brain with a number of electrodes that measure a neuron’s response (Hubel, 1957). The acquired spike trains from this process are processed statistically in order to extract information related to how neurons respond to a stimulus. Then a neuron can be classified by calculating its receptive field, i.e. the response associated to a conditioned stimulus, or using a discrimination threshold, i.e. the smallest change in a stimulus that can be detected by the neuron (Series et al., 2004).

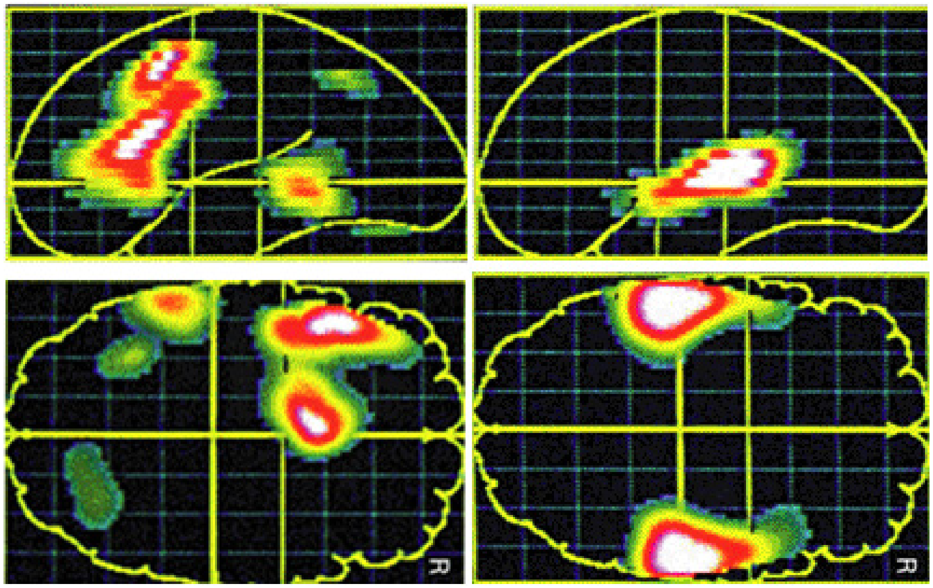


Fig. 2.3. PET image scans obtained during experiments studying human language. The four images demonstrate different regions that become active when subjects carry out language tasks. Image adopted from (Kandel et al., 2000).

The most common way to assign a cognitive role to this activity is by functional specialization: if a brain region is active during a specific cognitive process then it probably contributes to it in some way (Zeki and Watson, 1991). This concept is one of the dominant ways to interpret brain activity, and has produced a great amount of studies that localize brain function to specific cortical locations (Fig. 2.3).

For motor control, research has identified several networks that become active in the primate's brain, each dedicated to performing a specific function. As a result, various cognitive mechanisms that participate in motor control have been analyzed and studied extensively. The computational modeling approaches that have been developed based on these findings are reviewed in the next section.

2.2 Computational neuroanatomy of motor learning and control

In the previous section we have reviewed methods that permit brain activity to be measured, interpreted and modeled. In current section we look more closely into how relevant imaging results have been employed in order to develop biologically inspired computational models of motor control. Particularly, we focus on two categories of functions, imitation and motor execution, due to the fact that they are closely linked to the process of observational learning. Both processes are realized at a higher level in the brain, meaning that they employ a number of subliminary cognitive functions in order to be carried out. Motor execution pertains to the category of cognitive functions that enable a verterbrate agent to purposefully move its body (Schaal et al., 2007), while imitation is the cognitive process that enables it to integrate new motor knowledge based on its interaction with others (Billard, 2000; Schaal et al., 2003).

2.2.1 Computational modeling of motor control

From the perspective of computer science, motor control refers to the category of processes that transform the sensory inputs of an artificial agent into motor commands. Even though this appears to be a simple task, the intrinsic nonlinear dependencies imposed by the muscelotoskeltal system make it very complex (Synofzik et al., 2008). In addition, other difficulties that are associated with the control of high degrees of freedom bodies, such as the inaccuracies that are inherent in the perception of sensory signals and the continually changing properties of the environment, increase the complexity of the problem.

The fact that primates are able to exhibit robust motor skills effortlessly has steered the modeling community towards seeking solutions for the aforementioned problems in nature. In this context, cognitive neuroscientists suggest that motor control is a three stage process (Jeannerod, 1994): (a) at the highest level a path must be planned; (b) at the second level, the brain must compute the inverse kinematics for each position in this path; (c) finally, at the lowest level it must produce the appropriate sets of muscle forces in order to make the body move along the desired trajectory. In the computational modeling literature, these problems are solved using several subsidiary processes that can be summarized into four important motor control components: *(i)* primitives, *(ii)* internal models, *(iii)* state representation and *(iv)* sensorimotor integration. The computational implementations for each of these processes are reviewed and discussed in the following sections.

Primitives

From the perspective of a computational system, the control of a vertebrate's embodiment is a very complex problem. Most of its complexity rises from the fact that there exist inherent nonlinearities between the joints of the body. A direct consequence of these nonlinearities is that the effect of a control command depends on the state of the robot, and therefore it produces different outputs in different body postures. A direct solution to this problem is to form a state space of all the available motor input/output pairs and store them based on the possible configurations of the body. This however results in a high-dimensional and complex system, which becomes computationally implausible to solve in multi-part bodies (Gomi and Kawato, 1996).

To confront this difficulty, vertebrates have evolved their motor system to recruit their limbs' degrees of freedom into coordinated muscle patterns (Mussa-Ivaldi et al., 1994). This important convention reduces motor control to the regulation of a few basis motor behaviors, where each behavior is responsible for the coordination of one or more muscle groups. Such primitive based systems have been discovered in many vertebrate animals and in different forms (Grillner, 2006; Bizzi et al., 2008). In Macaque monkeys, neurophysiological experiments have revealed specific neurons that are correlated with the execution of basis motor behaviors (Rizzolatti et al., 1988). For example, their primary motor cortex is somatotopically organized, in a way that specific neurons respond to specific directions of the movement (Georgopoulos et al., 1988), and can be thought as encoding coordinated force patterns that move the hand towards a specific direction. Similar systems have also been observed in humans (Thoroughman and Shadmehr, 2000).

Computationally, the use of coordinated patterns of muscle activations provides an efficient framework (Fig. 2.4) for reducing the high-dimensionality that is inherent in multi-joint control (Ijspeert et al., 2002a; Ijspeert et al., 2002b). Primitives can serve as basis functions that help to solve the complex differential equations, by reducing the dimensionality of the computational problem to a tractable set of commands. Due to this, the literature has focused extensively on understanding the biological underpinnings of the motion of high degrees of freedom bodies using models of primitives (see Degallier and Ijspeert, 2010 for a recent review).

The basic intuition behind motor primitives is that they are a dynamical system that consists of two components. The first is a canonical system that drives every individual degree of freedom separately, while the second a system that couples and synchronizes different degrees of freedom together (Ijspeert et al., 2003). In vertebrates, bodies have evolved to perform two types of motions, discrete and oscillatory, each having its corresponding primitive model, attractors and limit cycles.

Limit cycles are phasic oscillatory movements. The first evidence of such system was provided by (Brown, 1912), which described a network of Central Pattern Generators able to generate rhythms in cats. Other experiments based on lampreys (Cohen and Wallen, 1980) and Salamanders (Delvolve et al., 1999) have revealed how a network can be designed in order to exhibit similar patterns of rhythmic activity. In (Kiehn et al., 1997) the authors have developed a limit cycle primitive model that can be used to produce oscillating movement patterns in robots.

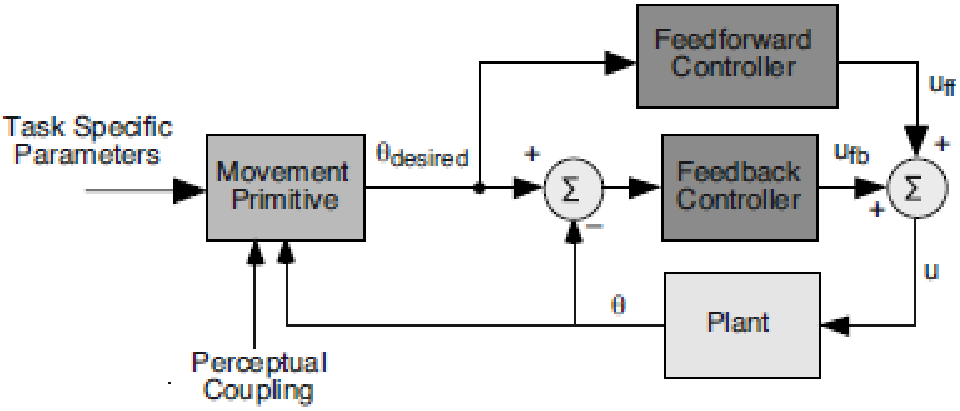


Fig 2.4. A schematic illustration of the role of a dynamic primitive model in a computational model of motor control. Image adopted from (Schaal et al., 2005).

In contrast, discrete primitives consist of a stable dynamical system that converges to a point attractor (Ijspeert et al., 2003). Biological evidence for the existence of discrete primitives has been found in frogs, which show how the spinal circuitry is structured in order to produce a modular movement (Bizzi et al., 2008 for a review). One of the first biologically inspired models of discrete primitives was suggested by Feldman (Feldman, 1966), and was based on creating a series of stable equilibrium points across the space of the robot's hand. Based on similar principles, Schaal (Schaal and Sternad, 1998) has presented a model of discrete primitives, which can orient the joints of the body in order to push it towards a specific posture. Their model inputs the joint of the hand and implements a function that generates the corresponding torque vector to achieve the desired pose (Fig. 2.5). More recent implementations include the FLETE (Bullock and Contreras-Vidal, 1993) and the VITE (Bullock and Grossberg, 1988a) models, which have been used in order to explain several different motor control phenomena based on self-organization (Bullock and Contreras-Vidal, 1993; Bullock, Contreras and Grossberg, 1992). The first model uses two neuron pools that specify the desired joint angle, in addition to a co-contraction signal that controls the joint stiffness. Control is achieved using two motor pools of interneurons (Baldissera et al., 1987; Hultborn et al; 1976) that inhibit the output in order to prevent saturation. In a similar manner, the VITE model (Bullock and Grossberg, 1988b) achieves limb control by using a neural network to change the lengths of agonist and antagonist muscles. Consequently the model does not change the trajectory of approach, but rather a desired final state of the system, encoded as a difference vector in the muscle of the body. Due to its simplicity, the model has been extended in various instances. In (Bullock and Grossberg, 1989) the authors used the VITE model to design a hybrid controller that can switch between joint and Cartesian space, while in (Gaudian and Grossberg, 1992) to model visually guided movements.

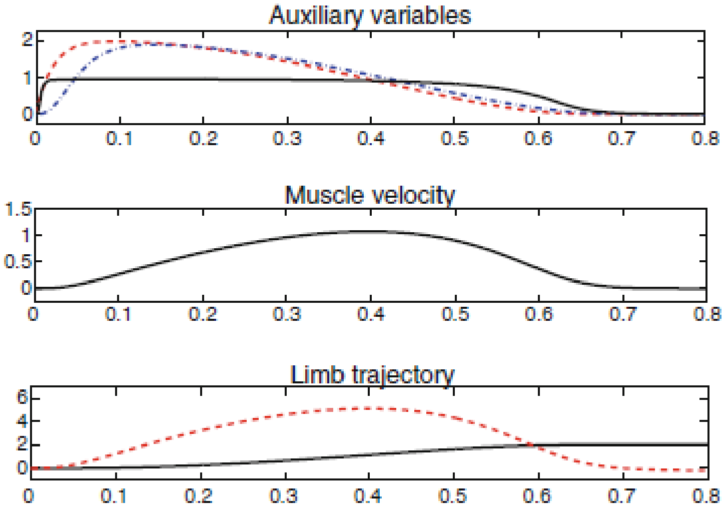


Fig. 2.5. The discrete trajectory as implemented by (Schaal et al., 2000). (a) activation of the model. (b) speed of the muscle in response to the activation of the primitive model. (c) the trajectory of the limb. In all three images the x axis corresponds to the timing of the model, while the y axis to the value of the plotted variable. Image adopted from (Degallier and Ijspeert, 2010).

To achieve flexible and stable control, it is quite common to combine the two primitive models, discrete and oscillatory, under the same dynamical system. For example in (Schaal et al., 2000) the authors present a dynamical system that can create a trajectory based on predefined properties and open task parameters. The discrete primitive in Schaal's system is based on the VITE model discussed above, while the rhythmic movement is triggered using a similar difference vector of amplitudes. Another primitive used to generate both types of movements is the one developed by Shoner and Santos (Shoner and Santos, 2001). In their work, the authors presented a model that can generate discrete and oscillatory movements using limit cycle attractors, able to resume its original trajectory even in cases where the limb position is changed during movement.

One important issue in defining a primitive model is its initialization. In humans, it is hypothesized that motor control is based on a simple model of primitives, that is synthesized using reinforcement learning (Doya, 1999). However, as Kawato has argued, proper construction of a primitive model can computationally become as complex, as solving for the actual equations of the inverse dynamics (Gomi and Kawato, 1996). To confront this problem, authors have used local learning methods in order to initialize their primitive models, the most popular of which being imitation learning (Schaal et al; 2007). In the next section we discuss how another component of the motor control system, the internal models, can be used to encode the representations required by primitives.

Internal models

In the previous section we discussed how motor control can be segregated into coordinated muscle patterns, in order to facilitate the movement of a complex muscelotosceletal system. In the central nervous system (CNS), this modular motion is encoded in the spinal circuits of vertebrates, i.e. at the end of the hierarchical system that is responsible for motor control (Mussa-Ivaldi et al., 1994). The higher cognitive centers of the brain are responsible for synthesizing them appropriately in order to generate a complex behavior.

This synthesis occurs on the basis of the task parameters at hand, and is strongly dependent on the state of the body. In this context, motor control can be regarded as a two stage process: *(i)* the representation of visual information to task goal parameters, such as a corresponding target position, and *(ii)* the consequent transformation of these parameters to torque commands (Gordon et al., 1994). To form such representations a computational agent must implement certain coordinate and state transformations that pertain to its body. In the CNS, these operations are carried out by a series of forward and inverse models, collectively identified by the robotic literature as internal models (Kawato, 1999; Tin and Poon, 2005).

For any movement, a controller needs to compute the force necessary for the muscle to make the required motion. This operation requires transforming the dynamical information of the arm, into the corresponding set of kinematic and dynamic parameters that are required to move it. In the brain, it has been suggested that these processes are being carried out by internal models that emerge as representations of the casual relationship between an action and its consequences (Fig. 2.6).

Representing such transformations however is a very computationally intensive process. Bellman (Bellman, 1957) has pointed out that the storage and computations required for a task is correlated to the dimension of its individual components. Since the body consists of 600 muscles, the number of possible motor representations scales up to an untractable number. Consequently, in the robotic literature, early solutions that pertain to maintaining table lookups (Atkeson, 1989) for all possible configurations of the body were abandoned for the sake of biologically inspired implementations of internal models.

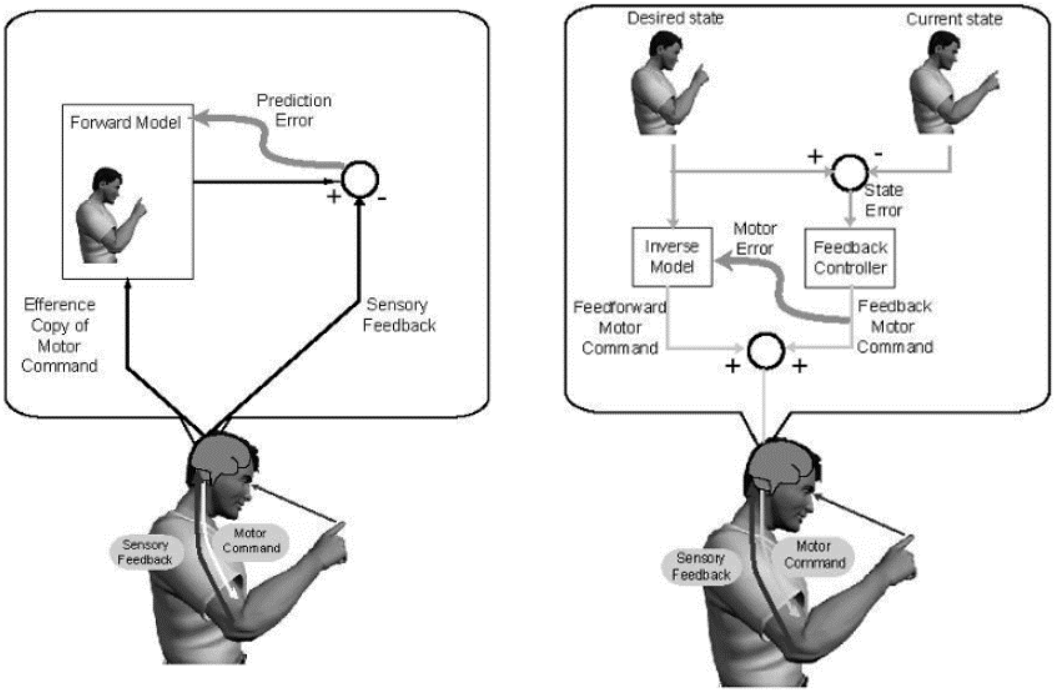


Fig. 2.6. Schematic illustration of the processes implemented in a forward (left) and inverse (right) model. Image adopted from (Arbib, 2003).

A forward model is a model that represents the casual relationship between the percepts of our body and the commands that must be exerted in order to control it. One of the problems in the computation of this transformation is that there exist inherent delays in the processing of the environment (Ito, 1970). For example feedback from vision can take up to 10ms to climb up the cognitive hierarchy of the occipital areas (Nowak and Bullier, 1997). In addition there are delays in the propagation of somatosensation signals, as well as inherent delays associated with motor control. Consequently, the role of the forward model is to learn to provide a future state estimate, by making a sophisticated prediction based on the agent's and environment's state (Kawato et al., 1987).

One of the main problems in this computation is that executed actions must be combined with consequences that are temporally delayed, usually separated in a timescale of tens of milliseconds (Ivry, 1996). A well-defined prediction model must provide the means to cope with the instabilities that arise from such delays in state estimation (Miall and Wolpert, 1996), and can be developed using an online feedback control method that will train it to predict the state variables, based on the history of motor commands and the control that is being elicited by the motor system (Ariff et al., 2002; Flanagan et al., 2003). A common practice to achieve this is by using supervised learning. Given a motor command, and the output of a network that predicts the consequences it has on the body of the agent, one can easily devise a supervised learning signal that will teach the network how to predict it (Wolpert et al., 1995).

The predictions made by a forward model are used by the motor system in order to identify the effects that a certain motor command has on the state of the agent. Each prediction, known as efference copy, is passed by the neural systems that implement the forward model into the corresponding systems of planning and action execution.

The second class of internal models that exists in the CNS is called inverse models (Wolpert et al., 1998). An inverse model is a system that can calculate the motor command that is required in order to achieve a desired state. In contrast to forward models which, as discussed above, can be learned relatively easy, inverse models require more complex computations, and are limited by the design of the motor control system. The role of an inverse model is to invert the operation of the control, and derive a motor command that will cause a desired state change, based on the state of the agent and the environment. In the literature, three main approaches have been used in order to teach such systems:

(i) Direct inverse modeling (Miller, 1987; Kuperstein, 1988).
(ii) Distal supervised learning (Jordan and Rumelhart, 1992).
(iii) Feedback error learning (Kawato, 1990).

In contrast to direct inverse modeling, distal supervised and feedback error learning use the error of the movement, in order to derive the errors in the motor command, and consequently can acquire a proper control model despite miscalculations in the movement.

Even in cases where the motor command is relatively simple however, training an inverse model still remains a difficult computational task. Most of the problems rise from the fact that there is an inherent delay in the execution of a motor command and its corresponding consequences. Therefore, it is difficult to derive a learning signal that will accurately predict the consequences caused by the body's movement. To compensate for this, researchers have suggested combining the operation of forward and inverse models (Fig. 2.7). The main intuition behind this approach is that the operation of an inverse model can be augmented using a forward model that is responsible for predicting the motor consequences. In the Macaque brain it has been suggested that a specific class of neurons in the ventral premotor area could provide such an interface between forward and inverse models, by converting observed actions into motor plans (Rizzolatti et al., 2001). In addition, neurophysiological evidence has shown that the cerebellum can be a possible site for implementing multiple forward and inverse models (Ito, 1970; Miall et al., 1993), where evidence demonstrate that the region constructs an inverse and forward model of the eye movement in order to learn simple reflexes (Shidara et al., 1993). This intuition has been employed in (Wolpert and Kawato, 1998), where it is demonstrated how combinations of forward and inverse models can carry out motor transformation tasks. A similar approach has been followed by Demiris (Demiris and Hayes, 2002) who suggested that a behavior can be learned by implementing the interactions between a forward and an inverse model using feedforward neural networks. In both approaches, the system produced was able to execute fast and coordinated movements. However their main drawback lies in the local encoding of behaviors, which means that an agent would require a distinct forward/inverse pair for each different behavior.

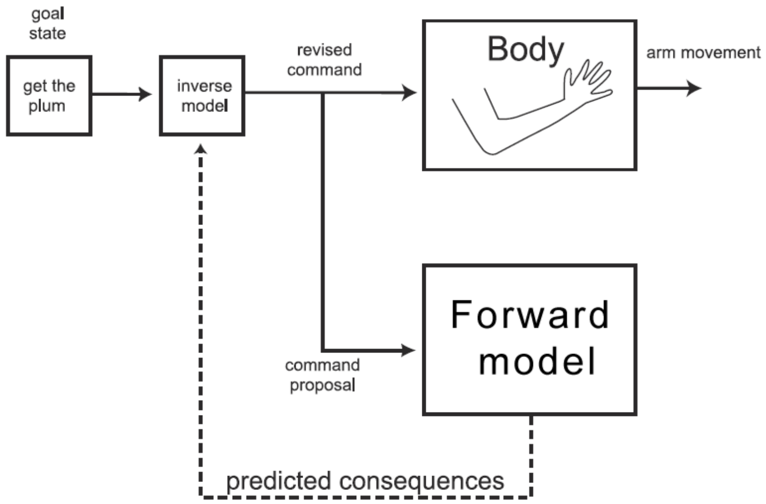


Fig. 2.7. A schematic illustration of how the processes implemented in a forward and an inverse model can be integrated together. The inverse model calculates the commands required to move the body, based on estimations derived by the forward model. Image adopted from (Churchland, 2002).

The ability of vertebrates to carry out complex movements is strongly dependent on the formation of accurate internal models (Wolpert and Ghahramani, 2000). Such cortical systems are responsible for adapting to the specifics of the executed movement, and produce appropriate joint torques that compensate for any unexpected events in the environment (Lackner and Dizio, 1994). This means that adaptation is a very crucial part in the design of an internal model (Shadmehr and Brashers-Krug, 1997). Psychophysical studies have shown that humans can adapt their motor control skills in other tasks (Conditt et al., 1997), directions of movements (Gandolfo et al., 1996), hand positions (Shadmehr and Moussavi, 2000), speeds (Goodbody and Wolpert, 1998) and visual motor environments (Krakauer et al., 2005).

Computationally, to achieve this adaptation, the internal model requires an appropriate representation of the state of the embodiment of the agent. To implement it, research has focused on using various neural codes, the most popular of which being population codes, because they can provide a spatial representation of the environmental state (Poggio, 1990). This type of fixed encoding provides a consistent representation of the environment (Pouget and Snyder, 2000) and therefore allows the system to adapt to any possible alterations in the motor control easily, by modifying the synaptic connections between specific neurons (Poggio and Bizzi, 2004). In (Josh and Maass, 2005), the authors demonstrate how a biologically inspired neural network can be trained to learn a motor control task using population codes.

In general, due to their ability to compensate for inherent problems in vertebrate movement, such as sensorimotor delays and feedback adaptation, internal models have been extensively used in order to model the state representation and estimation processes of motor control. More recently, research has also focused on how such models can be used to observe and interpret the movement of others (Wolpert et al., 2003; Mussa-Ivaldi and Bizzi, 2000; Lackner and Dizio, 1998). In the next section we discuss how such representations can be used by a planning component in order to facilitate complex, coordinated movement.

Planning and policy learning

Motor planning is the process during which the output of the motor control system is structured in order to achieve some behavioral goal. It is usually defined based on an external task, such as a target location or a trajectory that must be followed, and produces a deterministic sequence of movements that accomplishes it. Humans learn such task level representations during the early stages of the vertebrate motor development (Frith and Frith, 2003; Saxe et al., 2004).

In cognitive neuroscience, there is a clear distinction between trajectory planning and trajectory execution (Nelson, 1983; Bizzi et al., 1984; Flash and Logan, 1985). Planning involves the cognitive processes that resolve any redundancies in the body of the agent, and transform a task goal to the appropriate sequence of commands that must be followed in order to reach it (Uno et al., 1989), while trajectory execution is the utilization of these commands in order to move the body of the agent.

Depending on the underpinnings of the motor control system, there are four main approaches to planning. The first is reactive action selection, a stochastic policy that is usually implemented using architectures of actor-critics (Barto et al., 1983), and is based on the interactions of the agent with the environment. The second planning strategy is known as predictive action selection and is based on maximizing some value or cost function, by trying to increase the probability of receiving immediate or future rewards from the environment. The third action selection strategy represents each motor behavior as a discrete solution that carries out certain rewards when executed. The role of the computational model is to evaluate sequentially each alternative candidate, in order to find the one that can yield the most rewards. Finally, when actions are continuous, the action selection strategy can be implemented by finding the point where the derivative of a reward function is zero, in order to obtain the desired direction of movement.

Using the appropriate action selection strategy, a motor system can be formulated in terms of a learning model that must find the correct control policy in order to achieve the goals of the behavioral task. Mathematically, this problem can be defined as (Schaal et al.., 2003):

$$U = \pi(x, t, a) \quad (2.1)$$

where U is the output of the motor control system, π is the control policy that must be acquired by the robot, x is the robot's state, t is time and a is the parameterization of the computational model that is responsible for implementing the policy π. Equation (2.1) suggests that a policy should implement a direct mapping from the agent's state vector x, to a continuous vector U that controls its body. The search for this policy π is an optimization problem. It can be solved by assigning a certain cost function to the task, and using non-linear optimization methods in order to minimize it. This intuition has been extensively used in the framework of optimal control theory, where research has focused on suggesting solutions that minimize energy type quantities. For example, in (Hoft and Arbib, 1991) the authors describe how the kinematics of pointing movements can be explained by a cost function that minimizes the jerk, i.e. the first order derivative of the acceleration component. Using calculus methods, the authors were able to derive a fifth order polynomial that specifies the initial and final values of the acceleration of the movement. The minimum jerk hypothesis has also been investigated in (Flash and Hogan, 1985), while other researchers have focused on different aspects of energy functions, such as the minimum torque (Uno et al., 1989), the minimum metabolic energy (Alexander, 1997) and the minimum variance model (Harris and Wolpert, 1998).

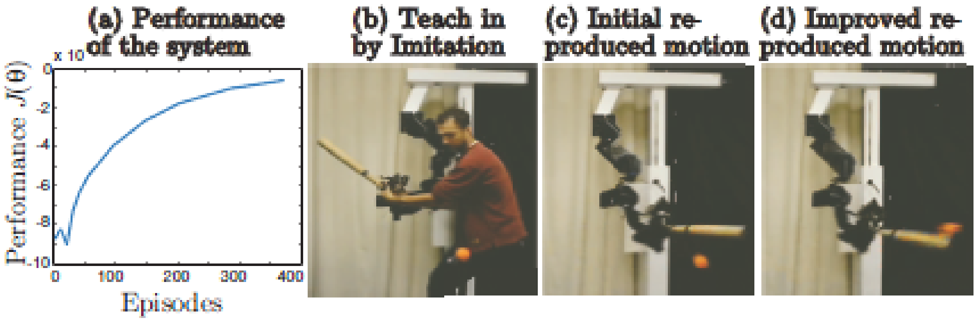

(a) Performance of the system
x 10
0
-2
-4
-6
-8
-10
Performance $J(\theta)$
0
100
200
300
400
Episodes
(b) Teach in by Imitation
(c) Initial re-produced motion
(d) Improved re-produced motion

Fig. 2.8. Learning motor control policies through imitation. The image illustrates a human demonstrating the task, and the corresponding behavior as executed by the robot. Image adopted from (Peters and Schaal, 2006).

Searching for appropriate policies is also compatible with the way primates learn. For example imitation learning can be regarded as a behavioral process during which policies are transferred from one conspecific to another by direct interaction (Fig. 2.8). This intuition has been exploited in robotics in order to develop models that can learn domain specific policies (Toussaint and Goeric, 2007), integrate new policies based on the demonstrator's representations (Peters and Schaal, 2007; Guenter et al., 2007) and learn through fast online algorithms (Bagnell et al., 2004). Other authors have explored how learning new policies can be achieved from observing demonstrating trajectories. In (Miyamoto and Kawato, 1998), the authors have developed a model that can extract an initial policy from a demonstrated movement, which is later refined through a self-learning process. Learning of new, non-autonomous, control policies has also been implemented using spline interpolation, where the model tries to fit a given trajectory into a spline, based on some via points which are temporally parameterized (Kawamura and Fukao, 1994). In a more biologically inspired sense, Billard and Mataric (Billard, 2000; Billard and Mataric, 2001) have developed a model where joint angular trajectories can be approximated by a policy that segments them for each joint movement, and demonstrated how it can be used to generate joint angle trajectories that were described by bell shaped velocity profiles.

Policy learning also becomes important in the context of primitives discussed in the previous section. If we consider that the motion of an agent is governed by the dynamical system described in eq. (2.1), then the search for an appropriate policy π can be determined by the repertoire of basis motions that are available to the agent. This research direction has been explored by many models. In (Ijspeert et al., 2003) the authors used policy search in order to enable robots to synthesize basic motor primitives into more complex behaviors. The same authors (Ijspeert et al., 2002a; Ijspert et al., 2002b) have suggested an alternative system to the spline based policy learning, based on autonomous dynamical learning and the use of motor primitives. In this context, searching for an appropriate policy is accomplished by rewarding all possible states, and identifying the one that will elicit the most rewards (Fig. 2.9). In such cases, planning is considered as a symbolic process, in which the representation of goals and percepts from the environment are combined in order to achieve the relevant task objectives (Lozano-Perez, 1982).

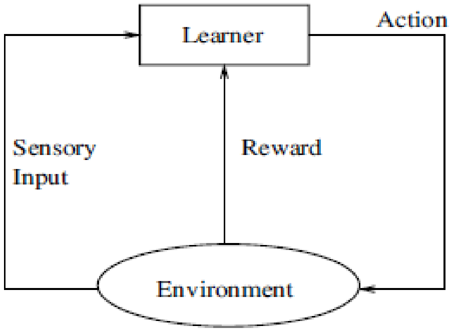


Fig. 2.9. A schematic representation of how rewards from the environment can be integrated within a computational model to solve action related problems. Image adopted from (Engelbrecht, 2007).

Most of the aforementioned action selection strategies are implemented using reinforcement learning (Lin, 1991). For example in (Atkeson and Schaal, 1997) the authors use this scheme in order to teach a robot a policy of the task dynamics, and demonstrate how goals can be used to learn task level policies of the movement based on reinforcement learning. Other models have employed different methods to facilitate learning of new policies, including supervised and competitive learning (Houk and Wise, 1995). In (Widrow and Smith, 1964) the authors used a supervised neural network in order to learn various tasks from human demonstration. The same intuition can be found in a number of works that implement direct policy learning, based on supervised signals derived from a robot teacher (Hayes and Demiris, 1994; Grudic and Lawrence, 1996; Sammut et al., 1992).

2.2.2 Computational modeling of motor learning

In the previous section we have reviewed the three main components that have been identified by computational modelers as the grounding elements of a motor control system. In the current section we discuss about the additional cognitive processes that are required in order to implement motor learning. Traditionally, in the computational modeling literature, motor learning is facilitated through imitation. Observational learning, the acquisition of novel skills only by observation, is a sub-category of imitation itself. For this reason, in the current section, we review topics that pertain to the design of imitation mechanisms in robotics, including embodiment, correspondence, and motor learning.

Action representation and the (embodiment) correspondence problem

Most contemporary theories in neuroscience claim that cognition is embodied, i.e. our thoughts and reasoning are tightly coupled and co-develop along with our body (Thelen and Smith, 1994). The importance of embodiment for cognitive development has been demonstrated by many research studies (Barsalou and Wiemer-Hastings, 2005; Damasio and Damasio, 1994), while extensive neuroscientific evidence have investigated how the link between the body and its perception can be formed and developed in biological agents (Tsakiris, 2010). This evidence has triggered extensive discussions on the important role that our body plays into forming our cognition (Synofzik et al., 2008; Tsakiris et al., 2007). The most simple definition of body ownership is that it is the perception of one's body (Gallagher, 2000). This is distinguished from the perception of an object mostly because the feeling of our body is always present (James, 1890).

In the context of motor control body representation has a very significant role, since our whole motor system is structured in order to confront the degeneracies that are imposed by our musculetoskeletal system. Moreover, to learn from a demonstrator, we must be able to match its body parts to ours. In the computational modeling literature this is known as the embodiment correspondence problem (Alissandrakis et al., 2002; Dautenhahn and Nehaniv, 2002). In practice, embodiment correspondence is a matching process (Fig. 2.10), which allows us to map the perceived state of the demonstrator, to the embodiment state of the agent (Byrne, 2003).

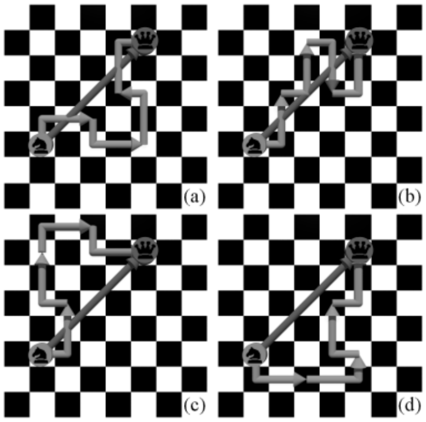


Fig. 2.10. Embodiment correspondence implemented based on the rules of a chess game. The different sequences that the knight can use to approach the queen pertain to different matching processes. In a similar manner, an embodied agent can exploit different strategies to match the body of the demonstrator. Image adopted from (Alissandrakis et al., 2002).

2.2 Computational neuroanatomy of motor learning and control

The cortical underpinnings of the correspondence problem have attracted a lot of attention among computational modelers and neuroscientists. For its solution two main theories have been proposed, *(i)* specialist and *(ii)* generalist. The first, proposes that there are specific functional correspondence mechanisms that are dedicated to imitation learning. For example in the Active Intermodal Matching model (Meltzoff, 2002), the authors suggest that the visual representation of a behavior is inherently converted to a representation that contains information about the body-part relationship. In contrast, generalist theories suggest that the correspondence problem is mediated by the same control mechanisms that have been developed for imitation. The Ideomotor Theory (Greenwald, 1970), the main advocate of this trend, puts forward the argument that the representations of body movements can be interpreted on the basis of the sensory feedback the agent receives.

The solution to the correspondence problem lies in the matching process between the perceived behavior and the agent's motor control system. In this context there are two fundamental theoretical components that must be understood. The first refers to the matching criterion that can be used when recognizing a certain behavior, while the second to the coordinate frame that the correspondence will take place (Meltzoff and Moore, 1994). For example if matching can only be specified in kinetic coordinates, the agent will not be able to perceive any relevant kinematic or dynamic properties of the movement. In (Miyamoto and Kawato, 1998) the authors explore how this frame transformation can be accomplished in movements specified in Cartesian coordinates, using the kinematics of the arm, which are defined in joint angle coordinates.

Alternatively, if the observer can only perceive movement related parameters, then the matching process is quite straightforward (Kawato, 1999). For example, when the trajectory is specified based on intermediate location points (Miyamoto et al., 1996), then the movement recognition system can be adapted using standard classification algorithms. Learning of dynamic parameters in this case can be accomplished using predictive forward models (Miall and Wolpert, 1996; Wolpert et al., 1998; Wolpert et al., 2003), in which the agent predicts the kinematic or dynamic parameters of a movement based on its own motor control system and the state of the demonstrator. Such intuition has been explored in (Demiris and Hayes, 2002), where the authors have used pairs of forward and inverse models in order to predict the properties of the observed movement. This bi-directional approach of behavior matching is compatible with recent neuroscientific evidence regarding the functioning of the mirror neuron system in primates (Rizzolatti et al., 1996), as well as contemporary theories of cognition such as the theory of mind reading (Gallese and Goldman, 1998).

To understand a demonstrated behavior the motor control system must implement a nonlinear mapping between the observed parameters of the demonstrator and the internal state of the observer. It is thus important for the teacher to be able to provide identifiable parameters, and for the student robot to be able to convert them to meaningful internal representations. This can be accomplished in a straightforward manner, if the student and the teacher share bodies that have the same kinematic and dynamic structure (Dautenhahn and Nehaniv, 2002). On the other hand, if the ability to observe directly the properties of the movement is not inherently encoded into a model, other methods can be used in order to uncover the hidden states of the observer (Arulampalam et al., 2002). In contrast, external coordinates, such as the acceleration of the end point of an arm are considered task-level representations. Task level representations are easier to perceive by observation and offer an implicit solution to the possible mismatch between the bodies of the demonstrator and the executor. However, they require an inherent knowledge of the motor control system, meaning that the agent must be able to convert the properties of the perceived movement from the external space into internal representations of its motor control system. This transformation can be accomplished with the use of inverse models (Bullock et al., 1993; D'Souza et al., 2001), such as the ones discussed in the previous section. In (Widrow and Smith, 1964) the authors train a neural network using such external coordinates, which are recorded directly from human behaviors, and demonstrate how they can be used in order to learn the pole balancing problem. Similar approaches have been used in various other computational models (Lin, 1991; Hayes and Demiris, 1994; Grudic and Lawrence, 1996), where the authors utilize movement representations defined in external task space in order to solve the embodiment correspondence.

The solution to the correspondence problem underpins the basis of motor perception. If computational agents can perceive and understand the motion of bodies of their conspecifics, then they can manage to learn from their observation. In the next section we discuss how such mechanisms can be utilized in order to build motor control systems that can integrate new knowledge.

Motor learning by imitation

Having discussed the underpinnings of motor control, in the current section we review the computational modeling literature that pertains to motor learning. In particular we will focus on biologically inspired approaches to imitation, because they are closely related to the process of observational learning. Most of the computational models in this field are based on neural networks, i.e. distributed computational systems that mimic the functioning of the brain in different levels of resolution (Jordan and Rumelhart, 1992). In the computational modeling community, imitation started attracting attention during the 80s, because it could provide an efficient framework for automating the manual programming that was required for the industrial robots used at the time (Fig. 2.11). Contemporary approaches to imitation are based on symbol learning (Dillman et al., 1995), while most of the models employ biologically inspired principles (Pook and Ballard, 1993; Arbib et al., 2000; Oztop and Arbib, 2002).

In primates, imitation is the main form of social learning because it facilitates the transfer of motor knowledge across agents by demonstration. The cognitive mechanisms that support imitation are developed in the early stages of the primate's life (Meltzoff and Moore, 1977), thus allowing the expansion of their motor system by integrating new knowledge (Bandura and Wood, 1989; Brass and Heyes, 2005; Iacoboni, 2005), and facilitating the social interaction of group members under the same context. In particular humans and apes are endowed with imitation abilities that are superior to other species (Bard and Russell, 1999; Whiten et al., 1996). For example, infants go through various developmental stages that pertain to imitation (Piaget, 1962), such as developing the ability to immediately replicate an action by observing it. In fully developed humans, neuroscientific evidence suggests that there is a common system for encoding action and perception (Viviani, 2002). This has caused the computational modeling community to redefine traditional views about cognitive imitation, held until recently (Wilson, 2001).

In the robotic literature, studies have extensively focused on the cortical process of imitation (Schaal, 1999; Breazeal and Scassellati, 2002; Dautenhahn and Nehaniv, 2002) mostly because it is a convenient way to reduce the size of the state-action space that must be explored by a computational agent during learning (Atkeson and Schaal, 1997). In (Meltzoff and Moore, 1995) the authors suggest that imitation can also be used as a bootstrapping process for communication, while recently, several cognitive, cultural and social theories that pertain to social interactions have been associated to imitation mechanisms (Montague et al., 2002; di Pellegrino et al., 1992; Frith and Frith, 1999; Piaget, 1951; Tomasello et al., 1993; Meltzoff and Moore, 1994; Byrne and Russon, 1998).

Despite the large attention, there is nowadays a little consensus on what is defined by imitation (Byrne and Russon, 1998; Heyes, 2002). In the literature, the most common form that can be found is true imitation (Tomasello, 1997), but there are also other forms such as contagion (Thorpe, 1963), response facilitation (Byrne, 1994), emulation (Nagell et al, 1993), deferred (Ito and Tani, 2004) and observational learning (Hourdakis et al., 2011).

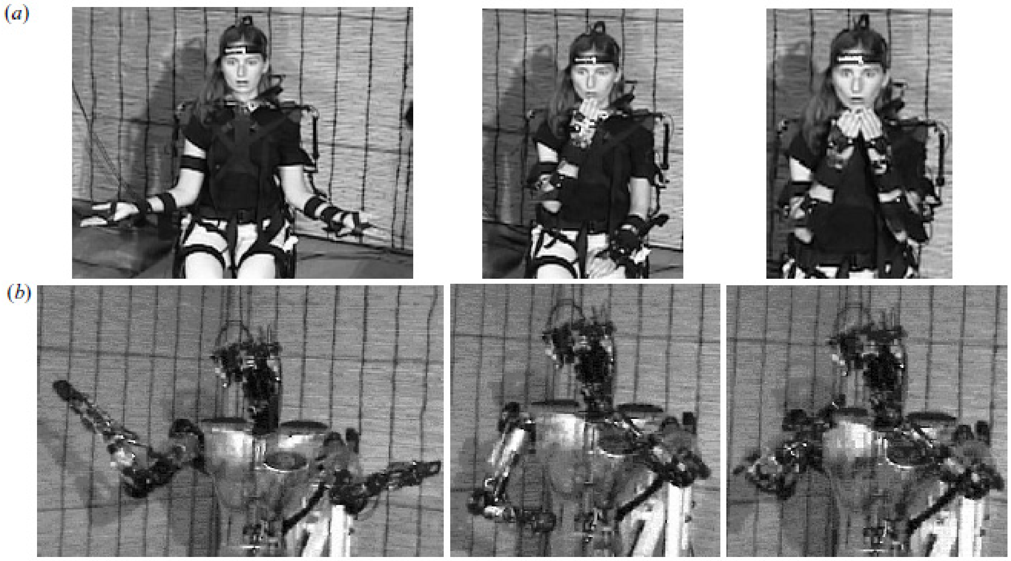


Fig. 2.11. Robot learning by demonstration. a. A human exhibiting a series of movements. b. A robot replicating those movements. Image adopted from (Schaal et al. 2003).

A general definition of imitation has been given in (Schaal et al., 2003), where the authors summarize the essential elements of an imitation system using the following equation:

$$F = g(z(t), u(t), t) \quad (2.2)$$

F corresponds to the loss function that must be minimized and is defined accumulatively based on the relevance between the motor behavior of the demonstrator and the teacher. The vectors z and u define the evolution of the internal state of the agent and external states of the environment respectively, while t denotes time. Equation (2.2) can be extended by adding supplementary terms that describe the deviations from the trajectory of the teacher and the demonstrator (Mataric and Pomplun, 1998; Nehaniv and Dautenhahn, 1999). To develop a basic conceptual system based on eq. (2.2) one must implement four important components:

(i) A visual system that can encode the action to be imitated.

(ii) Internal models that encode the sensory consequences and body states of the agent.

(iii) A component that encodes the goal of the imitated action.

(iv) A motor control system.

One of the main problems with implementing these components is that they are interrelated. For example, to convert the visual information into action one must take under consideration the underpinnings of the motor control system. In the case that this is implemented using movement primitives, as discussed in a previous section, one major question that must be confronted is how behavior matching can occur on the basis of primitive movements (Sutton et al., 1999; Sternad and Schaal, 1999) given that low level representations do not scale well when the body has many degrees of freedom (Schaal, 1999). In the following section we review the most prominent computational modeling approaches to imitation, and discuss how these issues are resolved.

Computational models of motor learning

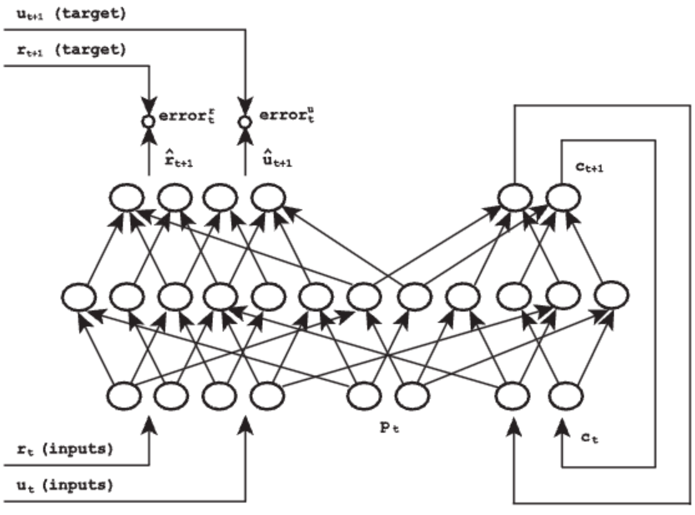
Due to the benefits that roboticists can gain from modeling imitation, several research groups have focused on developing methods for teaching robots by demonstration. Most of the approaches in this field have focused on neural network development, due to the fact that they can mimic, to a certain extent, the functioning of specific brain areas in the cerebral cortex (Vos and Scheepstra, 1993; Vos et al., 1997; Smeets et al., 1994; Willner et al., 1993). Although there exist interdisciplinary computational approaches that model imitation mechanisms (e.g. the use of Hidden Markov Models in Inamura et al., 2001), in the current section we focus only on biologically inspired neural networks because they are more related to the context of this thesis.

Most of the models in this field have been based on the neuroscientific finding of mirror neurons, a group of neurons in the premotor area of the cerebral cortex that fires both when a monkey executes and observes a specific behavior (Rizzolatti et al., 1996). Even though mirror neurons have been only associated with action recognition, many computational modelers extended their application to action generation as well, in order to suggest possible ways for implementing imitation. This assumption has been followed for many years in the literature, resulting in a series of models of mirror neurons that exhibit imitation based on an interpretation that was not fully compatible with the associated neuroscientific data. Despite this fact, research on this field has exhibited very competent models of motor learning, which we will review extensively in the current section.

In (Oztop et al., 2006) the authors have derived a basic taxonomy of the imitation models, based on the computational principles that have been used for their implementation. Consequently, they have categorized models depending on the learning method employed and whether they are consistent with the neurobiological findings that have been reported for the process. Even though, as we have discussed, most of the models that were developed in the early 00's were based on a misinterpretation of the neuroscientific data, they were able to exhibit one important property that is consistent with the way the brain of primates functions: action observation and action execution share a common coding system (Rizzolatti et al., 1996; Raos et al., 2007).

The first model we discuss (Tani et al., 2004) uses a recurrent neural network in order to learn spatio-temporal patterns, that are utilized as primitive behaviors (Fig. 2.12). The authors associate specific parametric bias vectors with basic motor patterns during a self-organizing process, that uses the same neurons for action generation and recognition. Each primitive is represented as a spatio-temporal pattern of activations, which is learned due to the recurrent connections that exist in the layers of the neural network. After training, the model is able to predict series of temporal sensorimotor patterns, and associate them with specific parametric bias vectors.

The latter are also used to recognize observed behaviors, and for this reason the authors have claimed that their architecture is consistent with the response properties of mirror neurons in the Macaque brain. One of the strong benefits of the approach is that all behaviors are stored distributively in the synapses of the recurrent neural network. However, even though the model provides an important basis for learning new behaviors as dynamical systems, it does not include any components for the planning or state representation processes. As a result, it can only be used to reproduce a spatiotemporal sequence of sensorimotor patterns, rather than act as a complete model of an imitation process.



(a) Learning Phase

Fig. 2.12. The RNNPB neural network during the learning phase, where the network learns new motor behaviors based on the parametric bias vectors. Image adopted from (Tani et al., 2004).

The next two imitation models we discuss (Demiris and Hayes, 2002; Haruno et al., 2001) have one important similarity. They implement a mechanism where new behaviors are learned based on the cooperation of a forward and an inverse model. Each recurrent controller inputs the target of a behavior and outputs the motor commands that are required in order to execute it. One very important assumption that both architectures make is that the demonstrator's joint angles are available to the agent at all times. The forward model is responsible for estimating the next state of the agent, which is directly fed to an inverse model in order to compute the necessary control commands. In the case of (Demiris and Hayes, 2002) each paired forward-inverse controller learns to output a confidence value that represents the extent to which the given behavior is similar to the one encoded. The same controller that is responsible for movement recognition is also responsible for movement execution, and based on that, the authors have claimed that their model is inspired by the function of mirror neurons. One drawback of the model is that behaviors are stored modularly, i.e. one requires a distinct forward-inverse controller for each new behavior. Consequently, the agent's storage capacity requirements scale up with the number of behaviors that are taught to the agent.

In the same line of approach, the MOSAIC model (Haruno et al., 2001) also employs internal models in order to implement the motor control component of the agent. Similarly to the (Demiris and Hayes, 2002) architecture, the authors suggest that a controller of forward-inverse model pairs can be used in order to encode a distributed representation of a behavior (Fig. 2.13).

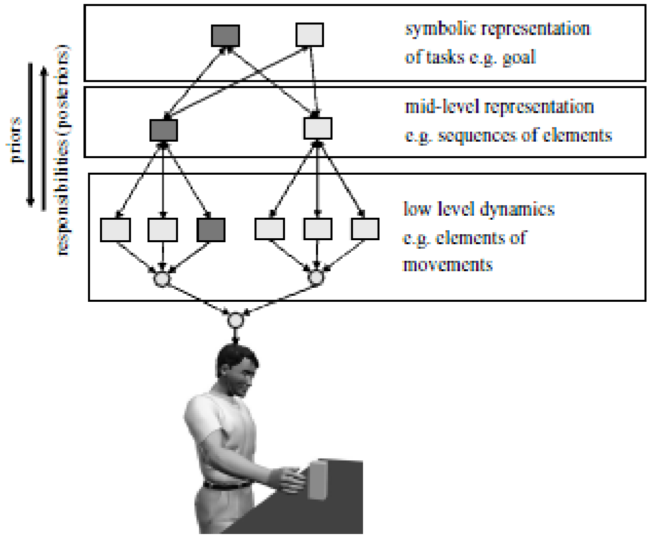


Fig. 2.13. The three hierarchical levels involved in the MOSAIC model. At the top level the goal tasks are represented, while at the second level, the motor sequence required to achieve these tasks. The bottom level involves the neural dynamics that produce the motor sequence. Image adopted from (Wolpert et al., 2003).

One of the strengths of the proposed model is that it is not associated with a specific learning method, but is rather a conceptual formalization of how a motor control hierarchy can implement learning by imitation. Due to this, various different algorithms have been adopted to the architecture, including gradient descent (Wolpert and Kawato, 1998) and expectation-maximization (Wolpert et al., 2003). Using the MOSAIC, the authors in (Doya et al., 2000) were able to implement an imitation system that is able to learn the task of swinging for a one-degree of freedom joint stick. One of the important components of the model is the use of responsibility signals, i.e. continuous values that represent predictions of the error of a controller when describing an observed action. Even though both models provide a consistent framework towards implementing learning of new behaviors by imitation, their claims of being a conceptual implementation of mirror neurons are not fully justified.

The fourth model that implements imitation based on biologically inspired approaches is the one developed by Borenstein and Rupin (Borenstein and Rupin, 2005). In their work, the authors follow a complete different line of thought from other related models, and investigate how evolutionary optimization can be used to prove the concept of mirror neurons on simulated agents. To accomplish this, the authors model the problem of observation and execution using an evolutionary framework, and show that when simulated agents are evolved to reproduce the output that is observed by a teacher, the fittest individuals develop an inherent tendency to use the same neurons for recognition and generation of an action. The use of genetic algorithms have been employed in several different motor control models in the literature, such as the ones in (Clif et al., 1993; Nolfi and Floreano, 2004), because they provide an attractive optimization framework to overcome the non-linearities that are inherent in motor control. One of the main drawbacks of these models is that the neural architectures that were used in order to implement the simulated agents were rather simplistic, i.e. they consisted of single feedforward neural networks. Moreover, the encoding of the teacher and agent states followed simple binary neuronal input, a convention that neglects the complexity of the actual biological input that is available to the mirror neurons in the Macaque monkey.

Using principles of associative learning (Hassoun, 1993), several authors have developed neural network imitation models of motor control. The benefits from this approach is that the developed neural networks are used as content addressable memories, which are computationally more efficient since they require only a partial input pattern in order to reconstruct the full representation encoded. The association in such architectures is performed usually between motor, visual and somatosensory stimuli. In (Elshaw et al., 2004) the authors use such an associative memory in order to correlate motor, language and visual representations. In their results, the authors claim that their model replicates the operation of mirror neurons, since the neurons in the hidden layer responded to both visual inputs and motor codes.

All the models in this category are based on the associative hypothesis of mirror neurons, which justifies mirror neuron development as the result of motor, visual and somatosensory events occurring concurrently when imitating. In (Billard and Hayes, 1999) the authors present the DRAMA architecture (Fig. 2.14), which can learn new spatio-temporal motor patterns using a recurrent neural network with Hebbian synapses. In (Kuniyoshi et al., 2003) the authors present an associative memory which can couple motor codes with visual information, while in (Oztop et al., 2005) the authors have developed a

Hopfield neural network that can generate different hand postures using Hebbian based synapses. One major drawback in these computational models is that they assume that the mirror neuron system is used during both action generation and action recognition, which as Rizzolatti pointed out is a false assumption (Rizzolatti and Craighero, 2004).

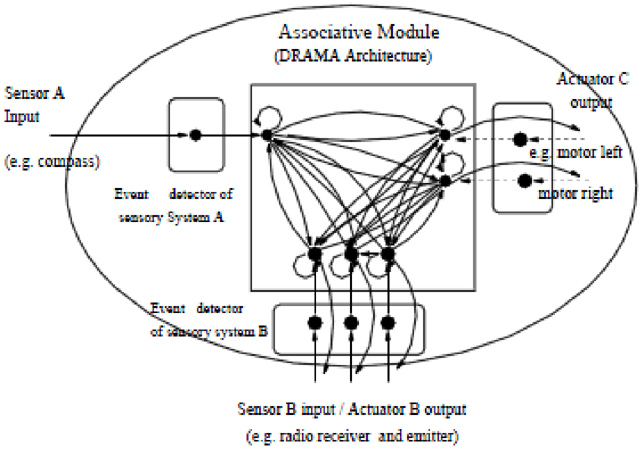


Fig. 2.14. A schematic illustration of learning new motor control processes using the DRAMA architecture. Image adopted from (Billard and Hayes, 1999).

The Mirror Neuron System (MNS, Oztop and Arbib, 2002) model is probably the most consistent, in terms of the cortical process of mirror neurons than any other model discussed up till now. This is because in their model, the authors make the important distinction between the mirror and canonical neurons in the prefrontal cortex, i.e. they discriminate between the cells that are responsible for action execution and action generation (Fig. 2.15).

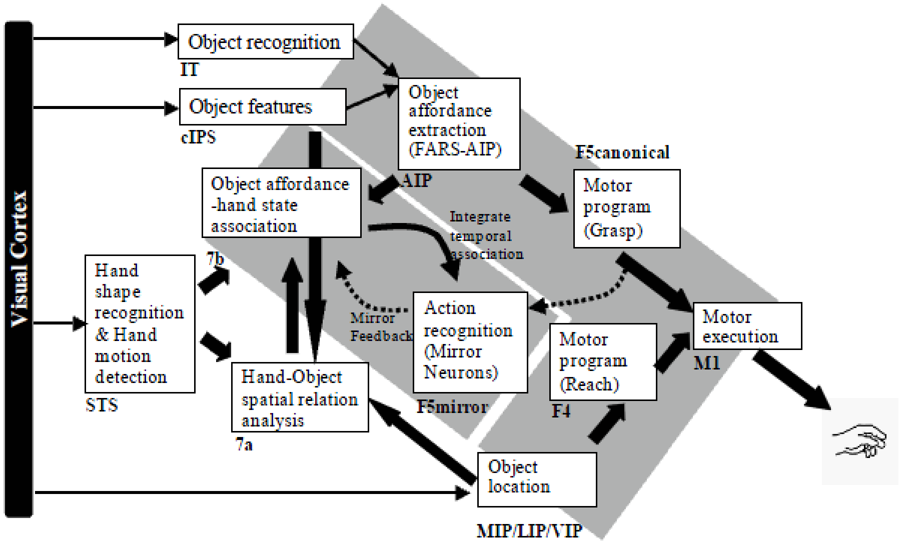

Visual Cortex
Object recognition
IT
Object features
cIPS
Object affordance extraction (FARS-AIP)
AIP
F5canonical
Motor program (Grasp)
Object affordance -hand state association
7b
Integrate temporal association
Hand shape recognition & Hand motion detection
STS
Mirror Feedback
Action recognition (Mirror Neurons)
F5mirror
Motor program (Reach)
F4
Motor execution
M1
Hand-Object spatial relation analysis
7a
Object location
MIP/LIP/VIP

Fig. 2.15. A schematic representation of the Mirror Neuron System developed by Arbib to model the cortical responses of mirror neurons during action recognition. Image adopted from (Arbib, 2004).

In addition, mirror neurons are developed strictly for action understanding, and not action execution. The intuition behind the model is based on the fact that mirror neurons have been hypothesized during infancy (Kohler et al., 2002), and consequently they must emerge through a developmental process that takes place in the early years of the primate's life. The first MNS model suggests that, the input that the infants receive during self-generated behaviors can be employed in order to reproduce the response properties of mirror neurons. The authors did not implement however a neural motor control component, while the remaining parts of the model have been designed conceptually, using schemas (Arbib, 1981). The model has generated several predictions regarding the function of mirror neurons, based on experiments that use different kinds of grasps and velocity profiles. In addition, it was extended by (Bonaiuto, et al., 2005) into a new version, which used recurrent neural networks, a more biologically plausible network than the backpropagation model that was employed by its predecessor. Moreover, the MNS2 model is based on additional neurophysiological experiments (Kohler et al., 2002), that demonstrate how sounds can be associated with motor codes in the auditory mirror neurons.

The final model we discuss is the one mostly related to the content of this thesis, as it uses the concept of mirror neurons in order to infer the mental state of other agents. To accomplish this, the Mental State Inference (MSI, Oztop et al., 2005) model is based on the assumption that the motor response of mirror neurons is only the utilization of a system with additional cognitive functions. Using visual feedback, the MSI model assigns a predictive role to mirror neurons, and postulates that their use is for understanding others' intentions (Oztop et al., 2005). The mirror neurons implement a forward prediction circuit in the model, which tries to infer the sensory consequences of the motor behaviors executed by the agent. However this assumption is not compatible with the view that mirror neurons are responsible for action recognition only.

All the aforementioned models are contemporary approaches to motor control and learning, inspired by findings from cognitive neuroscience. In the next chapter we review the cortical underpinnings of these processes in order to understand how they are facilitated in the brain of human and monkey primates. This will enable us to understand the neurophysiological details of observational learning in the brain, and derive computational agents that can support it.

Chapter 3

Cortical Underpinnings of Observational Learning

In the current chapter we review the neurophysiological studies that investigate the cortical properties of observational learning in the brain of human and Macaque primates. As already mentioned, the basic assumption that is employed throughout this thesis is that, the observation and execution of an action activates the same neural networks in the brain of the aforementioned species. This fact highlights one important property of the cerebral cortex during observation: *to understand a conspecific's behavior, primates reenact observed actions using their own motor control system (Savaki, 2010).* To shed more light into how this mechanism is facilitated in the brain, in the current chapter we analyze the regions that become active during execution and observation, and attempt to decipher their function using evidence from neuroscience. In addition, we examine the capacity to imitate in both species, in order to understand how the overlapping pathways facilitate observational learning.

3.1 Cortical regions that participate in action execution and action observation

The main hypothesis that inspired research work in this thesis is that the observation of an action activates the primary motor (MI) and primary somatosensory (SI) areas somatotopically (i.e. the forelimb representation for arm actions) in the same manner as the execution of the same action (Raos et al., 2004). One of the first findings of such common coding mechanism was reported in the Macaque ventral premotor Cortex (F5), where researchers have pinpointed the existence of neurons with mutli-modal properties (coined as mirror neurons, Fig. 3.1), able to respond to the observation and execution of goal directed movements (Di Pellegrino et al., 1992; Gallese et al., 1996; Rizzolatti et al., 1996). This simple mechanism was unique in its kind, and was interpreted by researchers as the basis for the recognition of others' actions (Rizzolatti and Fadiga, 1998; Jeannerod, 1994). In humans, neuroanatomical evidence suggests that the F5 area of the Macaque monkey is the precursor of Brodman area 44 (von Bonin and Bailey, 1947; Rizzolatti and Arbib, 1998), which also has an active involvement during observation.

Nowadays, neuroscientific studies have identified a plethora of regions, apart from F5 and B44, that participate in action observation. These have been collectively identified as the Mirror Neuron System (MNS, Rizzolatti and Craighero, 2004), even though there is still missing evidence as to whether all regions in the MNS contain mirror neurons. Nonetheless, analogous overlapping systems have been shown to exist in both human (Rizzolatti, 2005) and monkey primates (Raos et al., 2007), and as it is widely believed among neuroscientists they subserve the species' social capacities.

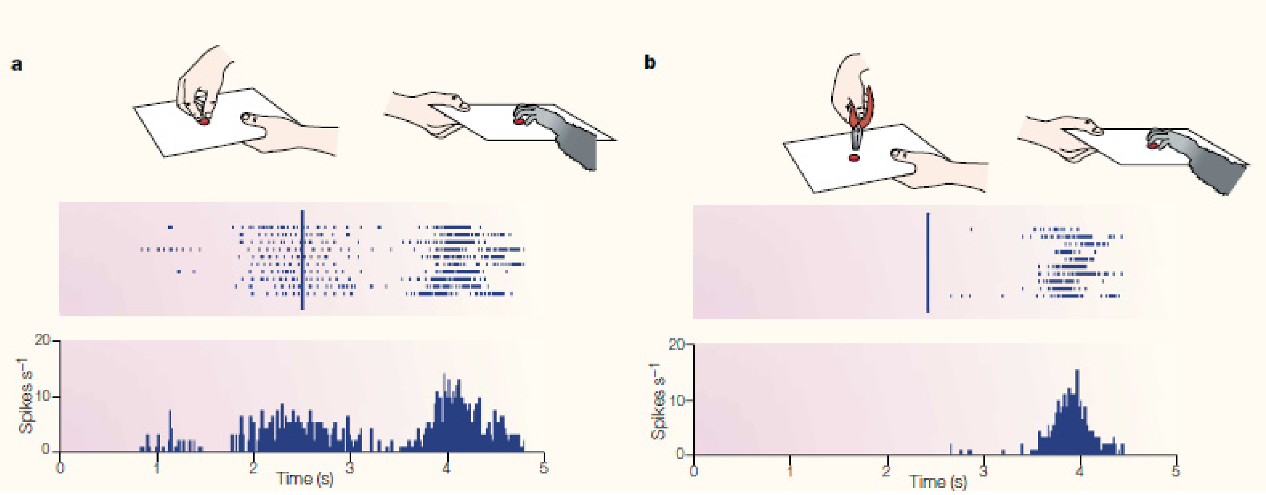


Fig. 3.1. Visual and motor responses of mirror neurons during grasping experiments. (a) The observation and execution of a grasp behavior elicits a firing if the experimenter uses his/her hand. Image adopted from (Rizzolatti et al., 2001).

Even though the existence of mirror neurons is considered to be an evidence of support of the simulation theory, most researchers nowadays suggest that our ability to understand others is underpinned by a pathway of activating regions instead of two areas (F5 and PfG) containing mirror neurons. Unlike the MNS, whose primal function is based on the activity of mirror neurons, new evidence suggest that there are plethora of regions being activated during action execution and action observation. For the sake of convenience, and to avoid confusion with the neurophysiological mechanism loosely coined as the MNS, in the remaining of the chapter, we shall name this mechanism Execution/Observation matching system (EOMS).

3.1.1 Macaque Monkeys

In monkeys, due to the ability to penetrate and record single neurons in the cerebral cortex, researchers were able to identify specific neurons that respond to the observation of others' actions. These, so called, mirror neurons have been discovered in premotor and parietal areas (Gallese et al., 1996), and are characterized by the fact that they discharge both when the monkey executes a goal-directed action and when it observes the same action being executed by a demonstrator (Fadiga et al., 2000).

More recently, ^{14}C-Deoxyglucose experiments have shown that the network of regions that is being activated in Macaques during observation extends further than the parieto-frontal regions and includes the primary motor and somatosensory cortices (Raos et al., 2004; Raos et al., 2007; Evangeliou et al., 2008; Kilintari et al., 2010). The same finding has also been confirmed by a recent meta-analysis of imaging studies, which showed that the action observation network expands bilaterally to both hemispheres, and far beyond the mirror neuron areas (Caspers et al,. 2010). These results suggest that when observing, Macaques make use of an extensive network of motor areas in order to understand what has been observed (Fig. 3.2). Quite interestingly this network includes the primary motor and somatosensory cortices.

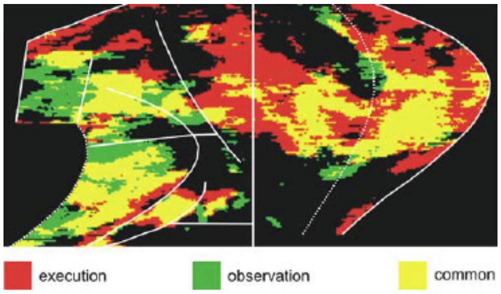


Fig. 3.2. Activations in the brain during an execution and observation task. Green areas mark the regions activated during observation, red areas the ones during execution, while with yellow the authors have marked the regions that are activated during both execution and observation. Image adopted from Raos et al., 2007).

3.1.2 Humans

The discovery of mirror neurons in monkeys has motivated researchers to investigate whether a similar overlapping system exists in humans (Rizzolatti, 2005). The first study to compare the observation of human hand actions and objects has reported the activation of the left inferior frontal gyrus, an area that contains motor representations of hand actions (Rizzolatti et al., 1996) and is homologous to the F5 area in the monkey (Petrides and Pandya, 1997). This finding gave impetus to numerous studies that investigated whether action processing in the human brain is characterized by the same cognitive properties found in Macaques (Buccino et al., 2004; Iacoboni et al., 1999). Nowadays, even though single cell recordings are not feasible, data from neurophysiological (Fadiga et al., 1995; Hari et al., 1998; Cochin et al., 1999; Brighina et al., 2000; Nishitani and Hari, 2000; Strafella and Paus, 2000; Sundara et al., 2001; Maeda et al., 2002) and neuroimaging (Decety et al., 1997; Grezes and Decety, 2002; Iacoboni et al., 1999; Astafiev et al., 2004) experiments were able to pinpoint the existence of a homologous overlapping network during execution and observation (Fig. 3.3).

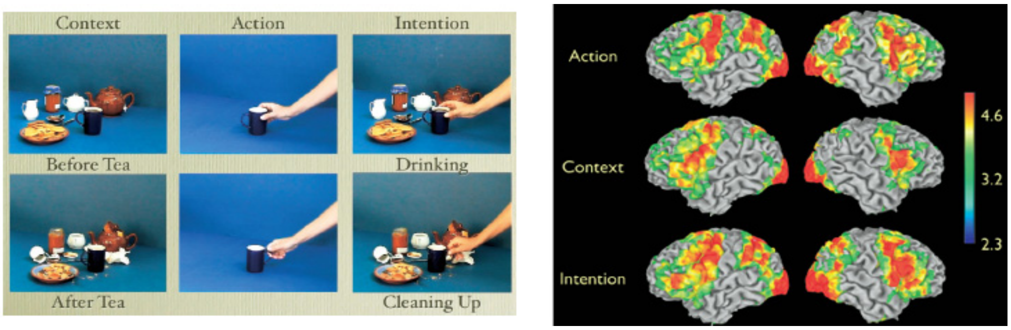


Fig. 3.3. Activation of brain regions during observation and execution experiments. Image adopted from (Iacoboni et al., 2005).

The above evidence is further complemented by neurophychological studies that confirm the role of the above mentioned regions to motor imagery. Sirigu (Sirigu et al., 1995) showed that subjects with degeneration in the primary motor cortex mentally simulate movements that are decelerated compared to their corresponding executed ones. The same author (Sirigu et al., 1996) has shown how lesions in the parietal lobe produce extended problems in our ability to imagine an observed action. In (Brass et al., 2000; Craighero et al., 2002) the authors have reported that when subjects are prompted by the observed movement they tend to respond faster, indicating a close link between the human observation and execution networks.

However, due to the restrictions in humans, it is not yet clear whether the above areas contain mirror neurons, or the extent to which the human EOMS system facilitates observation as in monkeys. Dinstein (Dinstein et al., 2008) has claimed that functional measures of brain activity are unable to conclude whether or not there are mirror neurons in these areas because they are limited to analyzing thousands of neurons, and therefore cannot provide evidence as to whether there are distinct cells that fire during observation and execution. Moreover, the same author has suggested that mirror neurons should show adaptation (i.e. a reduce in the energy output of a neuronal population after repeated stimulations, see Grill-Spector et al., 2006) in their measured responses, which was not confirmed at the time. This view however was recently dismissed due to findings that demonstrate a cross-modal (Kilner et al., 2009) and unimodal (Dinstein et al., 2007) adaptation in the neuronal populations of the human ventral premotor cortex. Nowadays, it is widely accepted among neuroscientists that humans have developed a formal imitation system that is extensively based on the function of the overlapping regions in the EOMS. In the next section we investigate the cognitive functions that are carried out in each of these regions, in order to understand how they contribute to the aforementioned processes.

3.1.3 Cognitive functions in the activated regions

The evidence discussed above suggests that motor execution and motor imagery activate the same regions and, to a certain extent, the same category of processes in the cerebral cortex (Jeannerod, 1994). In the current section we examine the neuronal correlates of the aforementioned regions, in order to understand the cognitive functions that they perform. In chapters 5 and 6 we will use this derivation to design the functionality of our computational agents. From the discussion in the previous section we have identified seven main regions that actively participate in observation and execution in monkeys and humans: (i) parietal cortex, (ii) primary motor cortex, (iii) premotor cortex, (iv) somatosensory, (v) visual, (vi) Basal Ganglia and (vii) SMA. The cognitive function that is performed in each region is briefly discussed in the following.

Parietal Lobe

The parietal cortex is a visual and somatosensory association area. In its inferior part, it contains neurons that are tuned to the object size and shape (Shikata et al., 1996). For example the Anterior Intraparietal area (AIP), located in IPL, contains neurons that are selective to (i) the shape properties of an object (visual neurons) and (ii) the execution of different grasp behaviors (motor neurons) (Fig. 3.4, Murata et al., 2000). These neurons also respond to goal directed actions, i.e. they fire as soon as the object to be grasped is presented (Sakata et al., 1995). Due to this, AIP is considered to play an important role in representing the affordances of a given object, through its connections with the F5 premotor area (Matelli et al, 1994), by associating the shape properties of an object with the motor behaviors that are required for manipulating it.

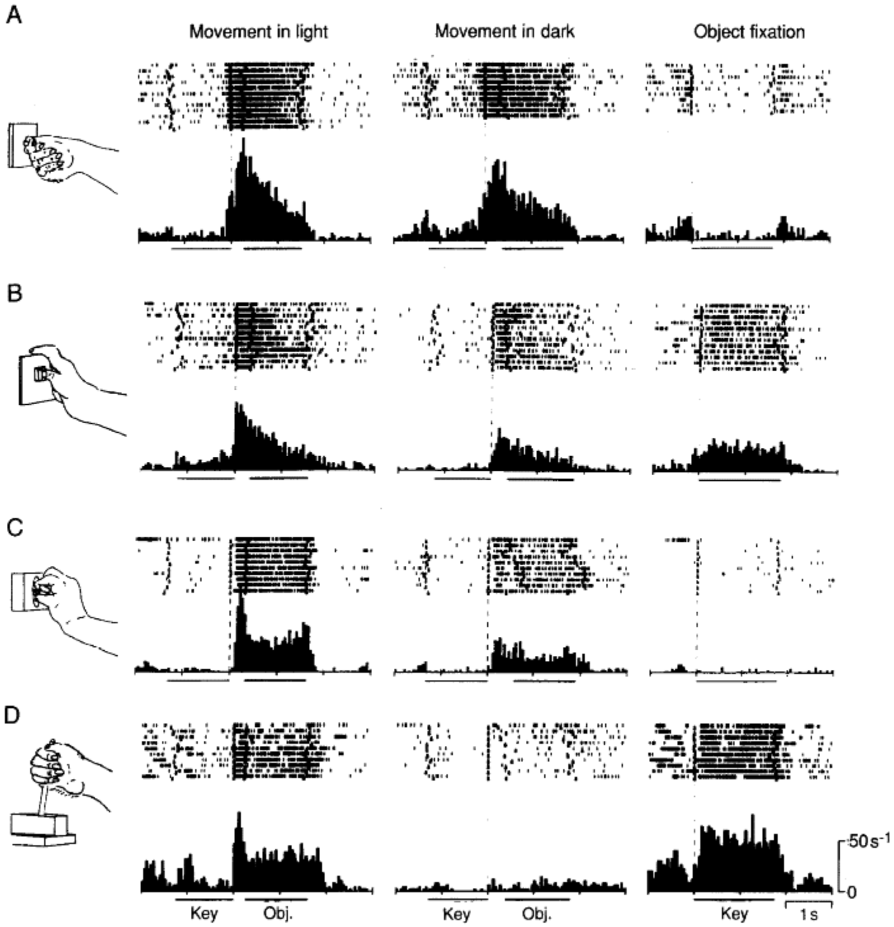


Fig. 3.4. Examples of different classes of neurons in the Anterior Intraparietal area (AIP) reported in (Murata et al., 2000). Image adopted from (Jeannerod et al., 1994).

The second area in the parietal lobe that was found to be activated during both observation and execution is the Superior Parietal cortex (SPL). It contains, among others, neurons that are selective to single and combined joint stimulations, i.e. their firing responses are correlated to the position of the agent's body parts (Sakata et al., 1973). For this reason SPL has been considered as a somatosensory association area, where various representations from the SI and other parietal regions are coupled to form a common neural code (Sakata et al., 1973). The region receives strong input from the somatosensory cortex and its neurons can be grouped into five different classes, based on the properties of their receptive fields: (i) directional skin, (ii) non-directional skin, (iii) single joint, (iv) combination of joints and (v) joint – skin (Sakata et al., 1973). Most of the somatosensation properties of the neurons in SPL are caused due to the projections from the primary somatosensory cortex (Jones and Powel, 1969). In its posterior part, where areas 5 and 7b reside, neurons have been associated with the perception of body form and extrapersonal space (Chow and Hutt, 1953; Crosby et al., 1962). Recent studies report that SPL also has an active role during the observation of a behavior (Evangeliou et al., 2008; Raos et al., 2007; Chaminade et al., 2002), supposedely by integrating proprioceptive and visual information, in order to form posterior beliefs about the position of the hand.

Premotor area

The premotor cortex is responsible for activating the correct motor behaviors in response to sensory stimuli, such as two dimensional patterns (Mitz et al., 1991), color (Halsband and Passingham, 1985), size and shape (Petrides, 1982). In its ventral part resides region F5, which is involved in the control and initiation of hand movements (Kurata and Tanji, 1986), and contains two classes of neurons that are related to motor control: canonical and mirror neurons.

Canonical neurons, when active, initiate a specific primitive behavior. They fire only when the object that is associated with an action is presented, but not to the presentation of the action itself (Murata et al., 1997), i.e. the neurons become active in the sight of objects that afford manipulation (Sakata et al., 1995). Jeannerod (Jeannerod et al., 1995) and Murata (Murata et al., 1997) interpreted this activity as encoding of segments of motor acts, otherwise referred to as motor schemas (Arbib, 1981). Motivated by these properties, Rizzolatti has suggested that F5 contains a vocabulary of motor acts (Rizzolatti et al., 1988), where some neurons encode general commands related to grasping, and others implicit information about the specifics of the grasp.

Mirror neurons become active when the primate executes or observes a goal-related motor act (Gallese et al., 1996; Rizzolatti et al., 1988). They are grouped into different functional classes, depending on whether they correlate with one or more elementary movement patterns, such as holding, tearing, manipulating and grasping. In monkeys, one of the distinct properties of these cells is that they will only respond to transive behaviors, i.e. ones that are coupled with objects in the environment (Rizzolatti, 1988). In addition, the firing properties in each class are not uniform: Some neurons will fire during the final steps of a grasping behavior, while others at the initial formation of the grasp.

Primary somatosensory cortex

The Primary Somatosensory cortex (SI) has four major cytoarchitectonic subdivisions, areas 3a, 3b, 1 and 2 (Kaas et al., 1979), each containing a complete representation of the contralateral part of the body (Fig. 3.5). Area 3a receives input from muscle and joint receptors (Krubitzer et al., 2004), and its neurons contain overlapping receptive fields for all fingers of the body. These representations are propagated to motor and parietal areas through the respective SI connections (Darian-Smith et al., 1993; Huffman, 2001).

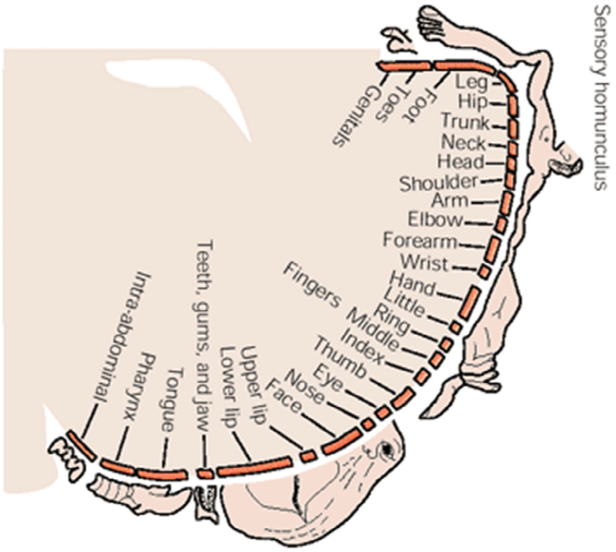


Fig. 3.5. Schematic illustration of the somatotopies existing in the primary somatosensory cortex. Each unique body modality has its own assigned population of neurons. Image adopted from (Kandel et al., 2000).

Individual areas in the SI are functionally distinct on the day of birth (Krubitzer et al., 2004), which suggests that its organization emerges developmentally in order to reflect the use of the hand. In the human EOMS, it has been proposed that the activation of the SI during action observation may be due to the anticipation of the somatensory consequenses of the seen actions (Savaki, 2010; Raos et al., 2007).

Primary motor cortex

The primary motor cortex is somatotopically organized in order to facilitate the control of different muscle groups (Penfield and Rasmussen, 1950). The receptive fields of its neurons overlap extensively within the area, and consequently the stimulation of specific neurons elicits the movement of multiple body muscles (Woosley et al., 1979).

Even though the region does not accept any direct visual sources, it does exhibit visual activity (Kwan et al., 1986; Riehle, 1991; Wannier et al., 1989), due to the indirect input from the premotor cortex (Barbas and Pandya, 1987; Kurata, 1991). Additional visual information is mediated through the connections from the Supplementary motor area, which in turn receives projections from area 7a in the parietal.

In general there is a large debate regarding the movement related properties in the MI (Evarts, 1968). Studies have shown that neurons are correlated with the force that is exerted by the body (Evarts, 1968), which is probably accommodated through the projections of the region to the spinal cord (Dum and Strick, 1991). MI cells also seem to be correlated with both magnitude (Schwartz, 1994) and direction (Georgopoulos et al., 1982) of the force of the hand. In addition, other parameters such as acceleration (Flament and Hore, 1988), movement initiation (Thach, 1978), target position (Crutcher and Alexander, 1990) and muscle co-activation (Humphrey and Reed, 1983) have all been associated with the activity of the MI neurons. Based on this evidence, Todorov (Todorov, 2000) was able to derive an almost linear approximation of the MI activity with the multi-joint kinematics in Cartesian coordinates as well as the end point force of the hand (Todorov, 2003).

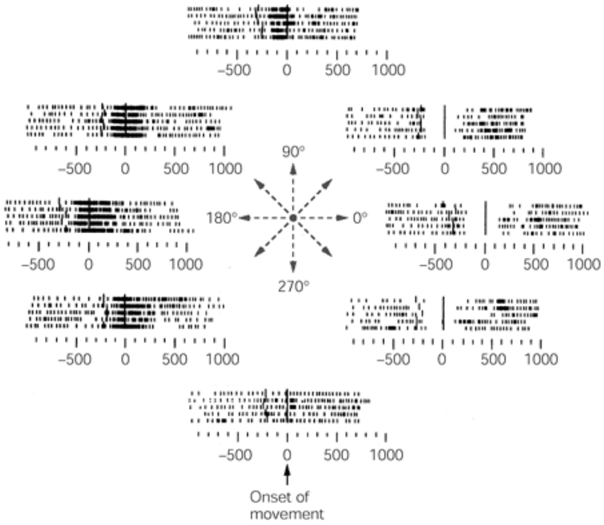


Fig. 3.6. Directional preference of the neurons in the primary motor cortex. The activity of individual neurons is correlated to the direction of the moving hand. Image adopted from (Kandel et al., 2000).

Due to the aforementioned properties the primary function of the MI region is believed to be the control of voluntary movements (Evarts, 1968). This is accomplished through neurons that are correlated with parameters of the motion, including force and direction of movement (Fig. 3.6, Georgopoulos et al., 1988; Caminiti et al., 1990), as well as the mechanics of the joints (Thach, 1978). The main contribution of the area in motor control is by dissociating the variables at each behavioral level (Saltzman, 1979), and exerting motor control commands through its connections with the spinal cord. The fact that MI neurons respond to discrete subsets of joint kinematics (Crutcher and Alexander, 1990; Scotts and Kalaska, 1997), as well as the fact that single neurons in the area move multiple arm muscles (Buys et al., 1986), in both monkeys (McKiernan et al., 1998) and humans (Colebatch et al., 1991) has led neuroscientists to believe that the behavioral role of the region is to control and synthesize movement primitives.

Supplementary motor area

The supplementary motor area (SMA) is also activated in monkeys during observation (area F6, preSMA, Raos et al., 2007) and contains neurons that produce a phasic response to visual signals of arm reaching movements (Kandel et al., 2000). In (Boecker et al., 1998) the authors reported that the activity of neurons in SMA is positively correlated with the increasing complexity of the movement that is being performed. This evidence, along with the dense projections it shares with the dorsal premotor cortex, has led neuroscientists to suggest that the former is mainly involved in the higher-order control of a behavior. This is accomplished by continually updating the related motor information and inhibiting various components of the motor control system (Shima et al., 1996). In general SMA participates in the planning of the executed behavior (Meltzoff and Gopnik, 1993) even though there is still not clear evidence as to how this is achieved.

Visual Streams

The cortical organization of the visual perception streams in primates is quite complex. Signals that originate from the photoreceptors at the retina travel through regions V1-V4 in order to reach higher order visual areas such as V5, STS, IPL, IP and STS. This integration occurs almost instantaneously, lasting at most 10ms (Nowak and Bullier, 1997), and depends on the feedforward connections between these regions (Thorpe et al., 1996). Visual processing occurs mainly at the higher order layers of the occipital and parietal lobes, which are responsible for assigning conceptual representations to these visual incentives (Fodor, 1982) through two different pathways: the ventral and the dorsal stream (Fig. 3.7, Ungerleider and Mishkin, 2000; Goodale and Milner, 1992; Ungerleider and Haxby, 1994).

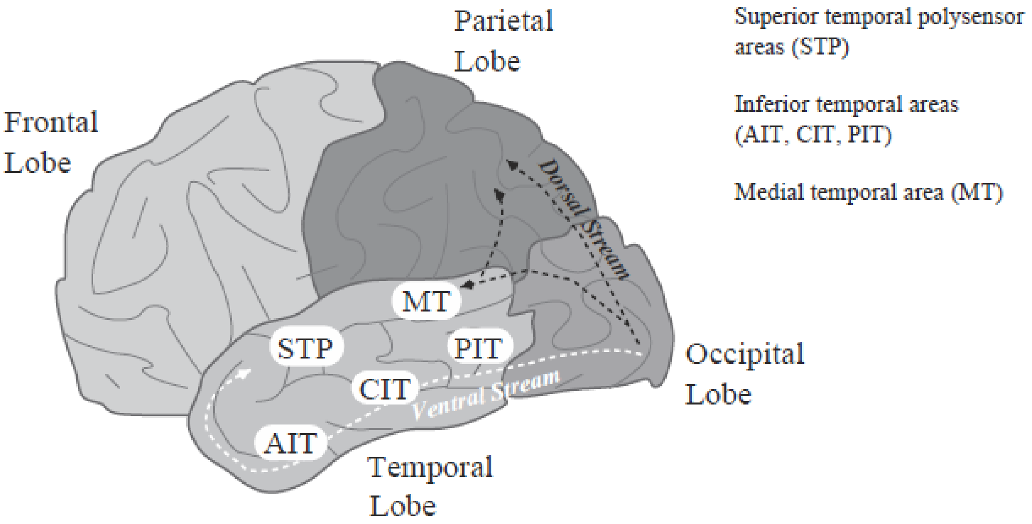

Parietal Lobe
Superior temporal polysensor areas (STP)
Inferior temporal areas (AIT, CIT, PIT)
Medial temporal area (MT)
Frontal Lobe
Dorsal Stream
MT
STP
PIT
CIT
Ventral Stream
AIT
Occipital Lobe
Temporal Lobe

Fig. 3.7. Ventral and dorsal pathways, which are involved in processing visual perception (Goodale and Milner, 1992). Image adopted from (Arbib, 2003).

The ventral stream defines the path from the visual cortex to the inferior temporal lobe. In its lower layers, the neurons are selective and specialized to specific stimuli such as shifts in stimulus position, size or illumination. These are integrated together in the inferotemporal cortex, where neurons show selectivity for concrete objects. In contrast, the dorsal stream projects to the parietal lobe and is responsible for categorizing the objects' visual features, by integrating visuo-motor transformations related to the hand – object interactions.

The visual recognition of body postures occurs in region Sulcus Temporalis Superior (STS). Neurons in the area become active when the monkey observes a conspecific performing a similar action but do not fire when the agent performs the same task (Perrett et al., 1989). For this reason, STS is believed to encode the observation of interactions between an object and an agent by exhibiting purely visual responses (Jellema et al., 2002). In humans, STS exhibits a greater activity during action observation, an effect that is attributed to the increased attention to the visual stimulus (Iacoboni, 2005). STS projects to the Ventral Intraparietal Area (VIP), which is also located in IPL, and contains neurons that are responsive to a moving stimulus, with a broader degree of tuning.

Spinal Cord

The spinal circuitry holds a hierarchical architecture of basic motor patterns (Nichols, 1994). It consists of networks of opposing channels, known as motor neuron pools, each responsible for the excitation or inhibition of particular muscle groups. During motor control, different neurons compete for the control of their related muscles (Heijst et al., 1998). In relevant experiments (Poggio and Girosi, 1990a), researchers have concluded that the spinal cord is made of discrete control modules that store limb postures as force fields (Poggio and Girosi, 1990b).

It is often usual to consider the spinal cord as a non-plastic region, however there have been several studies indicating evidence of self-organization in Sc (Wolpaw and Carp, 1993; Mendell, 1984). This is exhibited mostly at the initial stages of vertebrate motor development (van Heijst and Vos, 1997), for example during the initial weeks of the embryos, where spinal activity is more inherent (de Vries et al., 1982). Control is exerted by the Sc circuits based on its reciprocal connections with the primary motor cortex, where force field muscle primitives are stored (Mussa-Ivaldi et al., 1994). In addition, the spinal cord is somatotopically depressed during action observation, in contrast to its excitation during action execution (Stamos et al., 2010).

Basal Ganglia

The Basal Ganglia consist of a system of functionally distinct neural networks that accept information from different parts of the frontal lobe. Their function is to associate the stimulus from the environment with rewards, and due to their modular structure they are able to process concurrently different functions (Wilson, 1998). Presence of a reward results in the secretion of dopamine, where approximately 80% of the dopaminergic neurons exist in the Basal Ganglia.

Various neuroscientific studies have associated the Basal ganglia with learning goal directed behaviors (Graybiel, 1995; Miyachi et al., 1997). The dopamine neurons fire upon delivery of unexpected rewards (Shultz, 1998) or rewards that can be predicted (Shultz, 1998), and due to this property, the region is believed to play a crucial role in encoding the underpinnings of reinforcement learning (Montague et al., 1996).

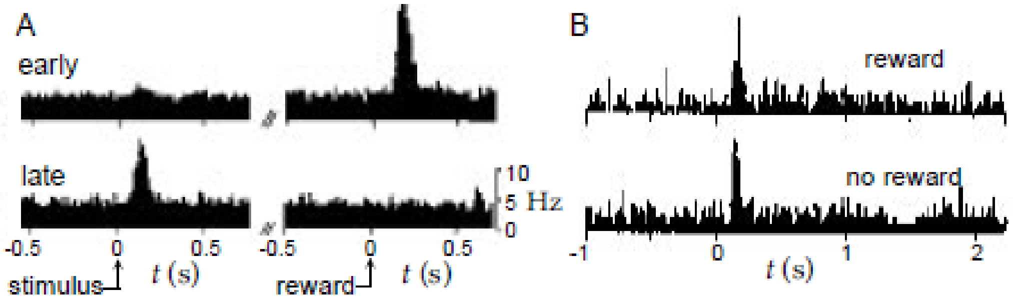


Fig. 3.8. Activity of the dopaminergic neurons in the Basal Ganglia. Images A and B show how the neurons can form their responses into predicting the rewards that will be elicit by the experimenter. Image adopted from (Dayan and Abbot, 2001).

The sub-circuit of the Basal Ganglia that pertains to motor control projects to the prefrontal lobe. During the execution of a behavior, the dopamine neurons display a phasic increase when the subject is given a reward or presented with a stimulus that predicts a future reward (Fig. 3.8). This phasic effect is important for inhibiting specific components of the movement. Additional studies suggest that the Basal Ganglia are mainly involved in learning motor behaviors (Graybiel, 1995) by predicting the reward outcomes of the executed actions.

3.2 Functional roles of the overlapping neural pathways

One of the first suggestions for the role of the MNS was that it facilitates imitation (Rizzolatti et al., 2001). However this view has received a lot of criticism due to the fact that adult monkeys are unable to exhibit imitation skills (Visalberghi and Fragaszy, 1990), and therefore the MNS could not have evolved to subserve this function. Due to this, cognitive neuroscientists nowadays suggest that the function accommodated by the mirror neurons is action understanding (Jeannerod, 1994).

A proposal about the function of the MNS, based on the mirror neurons properties, was put forward by Gallese and Goldman (Gallese and Goldman, 1998), which suggested that it is a system for *(i)* understanding actions based on their goals and *(ii)* deciphering the mental states of the observer. This mechanism requires an inverse process during which the goals that have been extracted by the observer will be recurrently propagated within the MNS circuitry (Wolpert et al., 2003). In cases where the observation of a behavior does not yield a match from the existing repertoire of known motor actions a new behavior must be generated.

In this context it is important to distinguish the types of behavior acquisition that can occur during imitation (Visalberghi and Fragaszy, 1990; Byrne, 2003; Heyes, 2001). One the one end, there is behavior matching, i.e. the ability to recognize an action that is already existent in the repertoire of the observer, while on the other, is the capacity to learn new motor sequences (Rizzolatti, 2004). The latter also facilitates behavior substitution, i.e. the ability to refine already known motor patterns from the repertoire of the agent.

As for learning by imitation, a process that requires complex mechanisms (Schaal et al., 2003), researchers have not yet concluded if it is supported by the initial MNS. For example, in Monkeys, the typical mirror neurons in the premotor and parietal cortices do not respond to similar biological movements if they are not associated with a target. This intransistive property of the cells suggests that mirror neurons do not fire in response to body displacements, but specifically to goal directed actions. The fact that mirror neurons are activated even in cases where the final parts of an action are not visible (Umilta et al., 2001), further supports the claim that the mirror neurons are associated only with action recognition.

In contrast to the claims that the MNS is a primary component behind these functions, more recently researchers were able to identify a more extensive overlap of activations during action execution and action observation (Savaki, 2010; Raos et al., 2004). This finding has given impetus to a more elaborate theory that explains how the aforementioned cognitive mechanisms can be facilitared in primates. The findings of overlapping pathways suggest that when we observe, we simulate the consequences of our actions up to the level of our somatosensory systems. Evidence that support this claim can be found in recent high resolution imaging studies, which show the activation of the primary motor and somatosensory areas during action observation (Raos et al., 2007; Raos et al., 2004; Kilintari et al., 2010;l Evangeliou et al., 2010; Savaki 2010). These additional regions are part of a more extended set of cortical areas, coined as the EOMS in this thesis, and provide evidence for the simulation theory and the ability of primates to reenact observed actions using their own cognitive system.

Based on the evidence we reviewed above, most regions in the EOMS are associated with handling motor control and its consequences. Because of this, the activation of the EOMS during the observation of a movement has led researchers to suggest that it is a mechanism evolved to subserve the mental simulation of others' behaviors (Savaki, 2010). In addition, a variety of other functions such as imitation (Carr et al., 2003; Gallese and Goldman, 1998), action understanding (Umilta et al., 2001; Gallese et al., 1996), intention attribution (Iacoboni et al., 2005), empathy (Wicker et al., 2003) and language (Rizzolatti and Arbib, 1998) have also been explained based on the properties of the EOMS.

Nowadays, the conceptual reasoning mechanism that is supported by the EOMS is one of the dominant theories for explaining humans' social abilities (Gallese et al., 2004). It is widely accepted among cognitive neuroscientists that mind reading is facilitated by a direct simulation of the observed behaviors, based on the functions of the regions in the overlapping pathways (Raos et al., 2007; Gallese and Goldman, 1998).

Chapter 4

Modeling Approach

In the previous section we visited the large body of literature that pertains to the cortical underpinnings of observational learning. To develop computational agents inspired from these findings, our first goal was to devise a proper framework so that the regional activation data reported for each cortical area can be integrated within a computational context, in a way that they will produce the appropriate behaviors in each agent.

In the current chapter we outline the underpinnings of a computational modeling methodology that was developed to assist this purpose. We first start the discussion by focusing on the problems that should be considered when suggesting a methodology for designing models of brain processes (section 4.1). Based on these principles, we continue to outline the core components of our proposed *pathway modeling methodology*, developed for designing biologically inspired models (section 4.2). In addition, we discuss contemporary approaches to modeling the brain with neural networks, as well as the mathematical formulation of the biologically inspired neuron model that was used throughout the development of the two computational agents that are considered in the current thesis (section 4.3). The chapter is concluded by presenting an analysis of the computational capabilities of the biologically inspired network that was employed for the development of the second agent, and the definition of a computational measure that quantifies its ability to classify different datasets (section 4.4).

4.1 Modeling the brain

Even though there is a vast amount of models developed based on knowledge from biology and neuroscience, there are very few methodologies for building such models. A well-defined computational model building methodology should be able to suggest specific means so that the information from biology can be embedded within a computational context, i.e. be able to define a cognitive behavior in engineering terms (Barto, 1991). To achieve this, it is important to:

(i) Suggest a model design approach that is based on the principles that underlie the brain functioning.
(ii) Provide means to integrate neuroscientific data within the model.

Both issues are discussed in this chapter. The former is discussed in section 4.2, in order to introduce our proposed method of computational modeling. The latter issue is treated in section 4.3, which focuses on how neuroscientific data from various imaging and recording studies can be utilized using biologically inspired neural networks.

4.1.1 Brain modeling methodologies

One of the most popular frameworks for brain modeling is Arbib's schema theory (Arbib 1992), which suggests that brain processes can be viewed as schemas, i.e. higher assemblies of regions responsible for specific functions within the model. Since it was introduced, the approach received increased attention (Arbib and Iberall, 1990; Arbib et al., 1990; Oztop and Arbib 2002; Tani et al., 2004), because it provides a method for integrating data from functional neuroimaging studies directly into computational models.

Even though Arbib's theory is a convenient way to model the functional activations of a cortical process up to the level of single neural networks, it cannot capture the dynamics of a behavior with respect to the whole model. Moreover, due to the interdependencies among cortical regions in the cerebral cortex (Phillips et al., 1984), functional localization, i.e. the identification of concrete functions in specific neuronal structures, is very difficult to demonstrate at the level of a schema.

Another theory that attempts to build a correspondence between natural and computational models is the parallel distributed processing framework (PDP, Rumelhart and McClelland, 1987; Rumelhart et al., 1986). PDP has been used to explain various effects of stimulus-stimulus and stimulus-response compatibility experiments (Zhang, Zhang et al. 1999; Erlhagen, Mukovskiy et al. 2006), by suggesting that multimodal information must be processed on an initial stage of the model, and combined with other information processing modules in order to achieve various cognitive tasks.

The ability of the methodology to develop a model based on parallel sub-processes can offer important computational benefits. However its strict commitment to parallel processing reduces the applicability of the method and makes it difficult to draw direct inferences between the generated models and the respective cognitive processes they are modeling.

Both aforementioned methodologies provide means to map the activation data reported for a region to specific engineering structures. However, in the cerebral cortex, it is not just the local regional activations that give rise to the richness of behaviors and cognitive functions, but also their functional interaction (Horwitz 1989). In this context, it is important to consider that an activation result reported for a particular region does not only portray the region's activity, but also the effect that other anatomically linked regions have, while projecting on it. In functional neuroimaging, such correlations can be extracted using structural equation modeling (McIntosh et al. 1994).

Computationally, task dependent correlations among interacting brain regions can suggest a pathway of associated functions. To portray them appropriately in a computational model, a methodology should not only be able to consider them, but also suggest ways to link them to behaviors. In the next section we describe our own approach to modeling cortical functions directly into computational models, and attempt to confront the aforementioned issues. The introduced methodology is employed extensively throughout this thesis, in order to develop artificial agents inspired by the cortical processes described in chapter 3.

4.2 Pathways and a modular approach to modeling

As already mentioned, both neurophysiological models of action observation involve a large number of cognitive functions in order to be carried out. Since the main goal of this thesis is to develop biologically inspired implementations of these processes, it is first important to device a theoretical framework that will enable the data reported in the aforementioned neuroimaging studies to be integrated within a computational context.

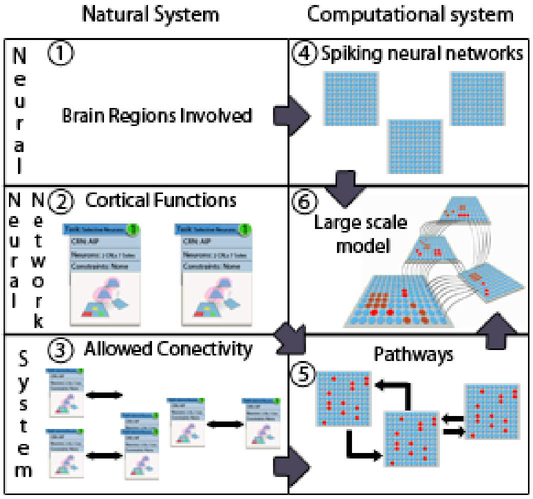


Fig. 4.1. Schematic illustration of how the pathway methodology can be used to develop biologically inspired computational models based on natural systems. The first three steps indicate the properties that the computational modeler must extract from the natural system, while steps 4-6 show how these are utilized by pathways in order to build a large-scale computational model of the process.

To accomplish this we focus on two principles that characterize the brain's interactions: functional organization and functional integration. The former pertains to the locality of functions within specific brain regions, while the latter, to the mutual activation of different functions in order to achieve a specific goal.

To model the above interactions we employ the concept of pathways from neuroscience, and formulate it within a computational context. Similarly to their biological counterparts, computational pathways are independent processing streams that perform a specific behavioral function. Each pathway consists of several computational regions, which cooperate in order to realize a specific modeled process (Fig. 4.1).

One region may belong to different pathways, however its involvement in any pathway means that it must contribute to it a specific function. To allow flexibility in this theoretical framework, we do not impose any particular constraints as to how each region must be modeled, even though in the current thesis, all regions are developed using biologically inspired neural networks. In particular, each pathway is characterized by two important properties:

1. The regions that participate in its processing.
2. The directionality of the flow of its information.

The first property allows a computational modeler to identify how the pathway will implement the function that it performs. Evidence for the regions that participate in a pathway can be derived based on neuroimaging methods such as fMRI, PET, MEG and ^{14}C-Deoxyglucose, which identify the activity of cytoarchitectonically segregated regions in the cerebral cortex during a specific cognitive function. Due to the fact that the pathway determines the connectivity between its regions, the activity of various different regions concurrently can give rise to phenomena of feature binding and association, which are believed to be the main form of learning in the cerebral cortex (Doya, 1999). For example most of the cortical synapses are based on associative rules, such as Hebbian plasticity. The main property of this associative learning scheme is that synapses will increase their strength when a pre-synaptic and a post-synaptic input correlate, and decrease otherwise (Tsumoto and Suda, 1979). As a result, neurons that belong to different regions within the same pathway will tend to increase their selectivity towards certain stimuli features, and form their response tuning based on the sensory experience of the agent (Blakemore and Cooper, 1970). This principle is employed throughout the regions of the computational agent we describe in chapter 5.

The second property specifies the basis upon which different pathways can interact together. To identify the directionality of the information one must look into how different regions are connected, and eventually how the signals will travel within a pathway, i.e. in a feedforward, recurrent or backpropagate manner. A direct consequence of this is that the modeler can easily identify the input/output conventions that are employed by each pathway, clearly discretizing between the inputs that a specific cognitive function requires, and the outputs that it will produce. This enables the integration of different processes together, into building a coherent, large-scale, computational model of a biological process based on the synthesis of autonomous/local functions.

The most important benefit from the use of computational pathways is modularity. As a principle, modularity is realized at all the levels of the processing hierarchy in the brain, from the low-level spinal systems to the higher-order association areas (Fodor, 1982). In (Plaut and Hinton, 1987) it is been suggested that the multiplicity of cognitive functions performed by the cerebral cortex implies that the brain uses modular architectures. The use of pathways can introduce the benefits from employing modular architectures directly into computational models.

From a computational perspective, modularity is a very important principle when it comes to designing large-scale biologically inspired agents. This is because complex processes often require many different tasks to be realized at once and therefore modularity can introduce important benefits to the model by discretizing dissociated functions, and explicitly defining which of the model's regions will be employed to accomplish them. Moreover, additional benefits from the use of modular structures include:

- Learning speed
  Decomposed functions are easier to learn than high dimensional ones.
- Generalization
  Networks that perform discrete functions can generalize better since they provide a clearer, distinct representation of the input/output conventions that are used to train them.
- Coping with errors
  Modularity can confront traditional computational modeling problems such as cross-talk, which arise from the increased interdependencies in distributed architectures.

4.3 Modeling the brain with biologically inspired neural networks

Having established a method for describing the functional interactions among regions, in the current section, we discuss how each cortical area can be modeled. In chapter 3 we have reviewed how imaging techniques are relying on various measures of neuronal activity, such as blood flow and glucose consumption, in order to understand the cortical underpinnings behind the brain's function. Based on these findings, we discuss how this data can be modeled using Neural Networks, in order to capture detailed aspects of a cognitive process, and map them into specific engineering structures.

In the current section we only focus on the general modeling principles that were employed during the development of the two agents, and describe the neural network and neuron models that were used. In later chapters (chapters 5, 6) we show how these biologically inspired tools were used to develop the appropriate representations in the two agents.

4.3.1 Neural Networks

Early explorations of the brain's function led to the theory of representation, i.e. the view that cognition develops based on static, segregated symbols (Newell and Simon, 1976). Using these symbols, our brains have co-evolved in order to produce meaningful communication (Deacon, 1997). Nowadays, most brain modeling research uses neural networks as a building block to construct such representations. Even though a neural network has no greater relevance to a cortical process than any other method (Miall, 1989), it is usually preferred because it shares similar structural properties with the brain:

- *(i)* It consists of a large number of densely interconnected, processing units.
- *(ii)* It represents knowledge distributively, based on the collective activation of its neurons.
- *(iii)* It violates the same sepparability assumptions of information processing (Palmer and Kimchi, 1986) as the cortical regions in the brain.

Due to the above reasons, in the computational modeling literature, neural networks are usually treated as representations of a specific hypothesis, about the structure and function of a region in the nervous system (Nordlie et al., 2009).

The most important grounding unit of a neural network is the neuron. It is the elementary computational building block, which performs a single operation and controls the nature of the neural code that will be exhibited by the network. In (Maass, 1997), neural networks are categorized into 1st, 2nd and 3rd generation, depending on the neuron type they use. The first generation networks consist of the McCullogh-Pitts model, a conceptually simple neuron that performs only a single binary operation. In contrast to the first generation networks, neurons in the second generation produce a continuous, analog value as output. Due to this property they have been associated with a number of learning rules, such as reinforcement learning and backpropagation (Zipser, and Andersen, 1988).

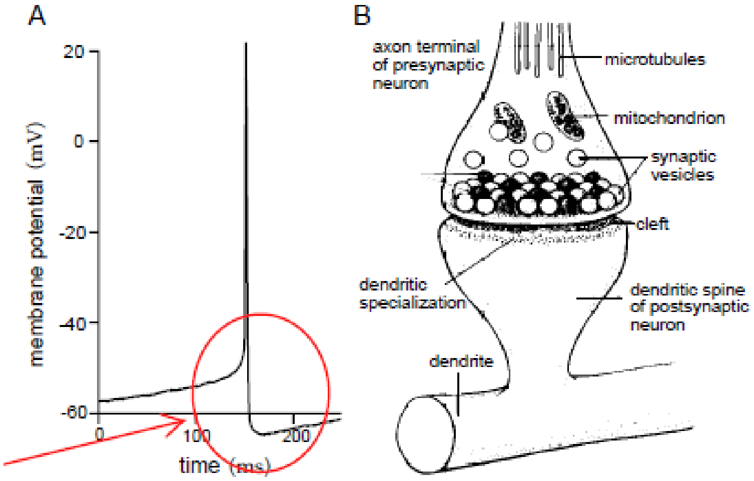


Fig. 4.2. The action potentials emitted by a spiking neuron model the membrane dynamics in the actual cortical cell. Image adopted from (Dayan and Abbott, 2001).

The third generation of neural networks is more consistent with the function of the cortical cells, than any of its predecessors (Fig. 4.2). The only output produced by these neurons is in the form of a spike, i.e. a binary pulse that has a temporal dimension. The main advantage of spiking neuron models over formal neurons is that the latter use a weighted sum of the input, and therefore cannot convey any temporal information, which is important when modeling cortical processes.

Relevance of the spiking neuron model to cortical cells

In contrast to other neuron models, spiking neurons model the electric effects that are produced due to the distribution of ions in the membrane of the actual cortical cells. This class of models has its origins on the first Hodgkin and Huxley model (Hodgkin and Huxley, 1952), and was designed to describe the temporal change of the sodium, potassium and leak currents during axonal transmission. The model has been successfully applied in a number of different experimental situations (Jack et al., 1975), and was recently used to describe the underpinnings of spike generation in cortical neurons (Ekeberg et al., 1991; Bush and Douglas, 1991).

Despite its close biological relevance, Hodgkin and Huxley neurons have a considerable computational overhead, due to the high dimensionality and nonlinearity of the differential equations used. To compensate for this, researchers have proposed variants of the initial model (Nagumo et al., 1962; Abbot and Kepler, 1990; FitzHugh, 1961) that make different types of simplification assumptions. The most famous reduction is the one proposed by Stein (Stein, 1967). In the so-called Leaky Integrate and Fire (LIF) model, the effects of the potential difference between the cell and its surroundings, caused by the concentration of the ion channels in the membrane, are reduced to a single analog value, the membrane potential (Fig. 4.3).

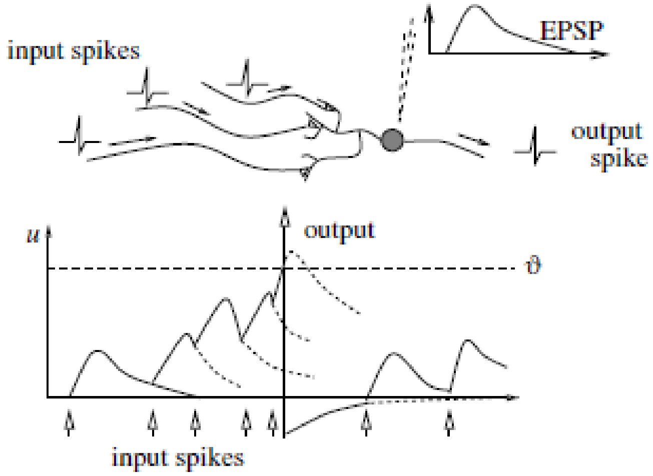


Fig. 4.3. How the fluctuations in the membrane potential cause a neuron to fire a spike-after potential. Image adopted from (Gerstner and Kistler, 2002).

The neuron is characterized by supplementary properties that modulate its firing response:

(i) The refractory period models the timing interval after the emition of an action potential, during which the membrane is not accessible.

(ii) The threshold value characterizes the limit of ion concentration that the soma must exceed before emitting a spike.

(iii) The resting potential, which describes the hyperpolarization phase that the neuron undergoes after the emission of a spike.

The LIF model is the most popular amongst researchers, because it holds a delicate balance between modeling resolution and neuronal consistency. In (Abbott and Kepler, 1990) the authors demonstrate how the Hodgkin and Huxley model can be reduced to a LIF model, in the limit where the timing properties of the membrane are the dominant time scale.

More importantly, depending on the input and the configuration of the neuronal parameters, the model can exhibit a variety of neural codes (Fig. 4.4). For example networks of LIF neurons, with a threshold value above the average intracellular current input, can act as coincidence detectors (Kempter et al., 1998), implementing a variant of radial basis function units in the temporal (Hopfield, 1995) domain (in chapter 5 we explore this principle in order to bind different motor and visual representations together). In (Eliasmith and Anderson, 2003) the authors describe how an elementary network of spiking neuronal networks can perform various mathematical operations such as addition of scalar values and vector representation. In (Christodoulou et al, 1992) the authors demonstrate how the neuron can reproduce the spike irregularity of biological trains, by changing the reset mechanism of the membrane potential.

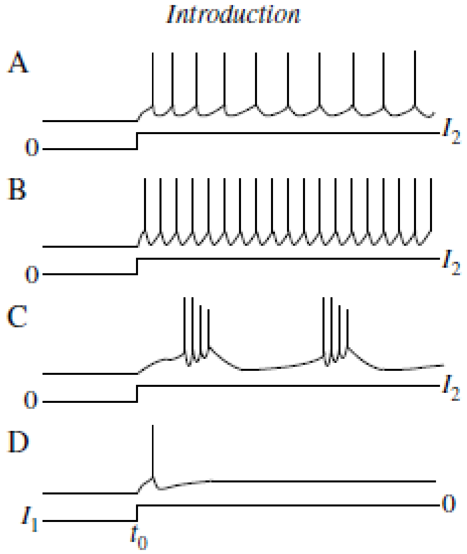


Fig. 4.4. Different neural responses exhibited by a spiking neuron. A. Irregular firings, B. Regular firings C. Spike bursts and D. Latency code. Image adopted from (Gerstner and Kistler, 2002).

Perhaps the most interesting aspect of the spiking neuron is the temporal integration of the membrane potential. This property enables the neuron to form a short-term memory of its inputs. For this reason, the most common use of spiking neural networks is in circuits that transform their inputs into a spatio-temporal pattern of neuronal activations. Popular examples of such networks include the model of synfire chains (Abeles, 1991) and the Liquid State Machine (LSM, Maass et al., 2003), which are used in the implementation of the second agent and are reviewed later on in section 4.3.3. A detailed review about the properties of the LIF model can be found in (Jack et al., 1975).

It is important to note however, that real cells have a larger variety of ion channels, than the neurons in the giant axon squid (Manwani and Koch, 1999), and therefore can exhibit a richer set of properties. Even though there are compartmental models that can compensate for the spatial dimension of the distribution of ions in the membrane's potential, they require a much larger computational overhead and are difficult to be incorporated into large scale computational models.

4.3.2 The Leaky Integrate and Fire neuron model

For the implementation of the neural networks in the two agents we employ the aforementioned spiking neuron model (Maass, 1997), which has a non-continuous, non-linear synaptic response, and resembles to a high extent the behavior of biological neurons. Spiking neurons have been widely used in modeling biological brain regions (Todorov, 2000; Kempter et al., 1998) and brain functions (Christodoulou et al., 1992) since they can approximate any continuous function (Maass, 1997), capture the temporal properties of their pre-synaptic inputs and exhibit short-term memory effects (Bugmann 1997). Additionally, they can be coupled with a large variety of associative (Song et al., 2000; Gerstner and Kistler, 2002) and reinforcement learning rules (Baras and Meir, 2007). Several variations of the spiking neuron have been proposed (Jack et al., 1975; Stuart and Sakmann 1994; Abbott and Kepler 1990), ranging from anatomically consistent to computationally oriented (Gerstner 1998), all having their basis on the initial Hodgkin-Huxley model (Hodgkin and Huxley, 1952; see also Cronin, 1987 for a mathematical analysis).

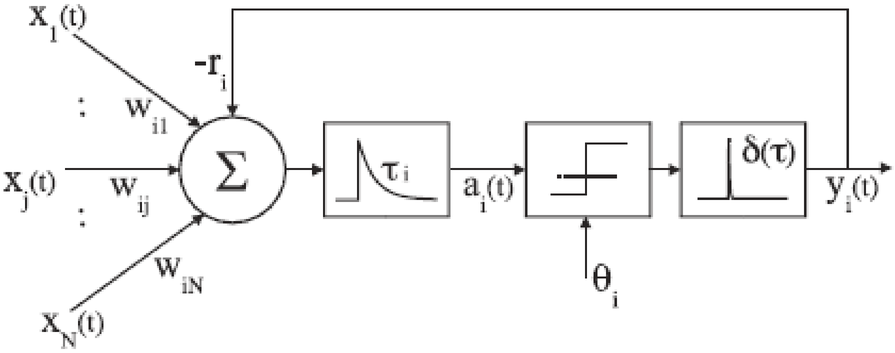


Fig. 4.5. Schematic representation of the computations (input transformation, spike generation) carried out by a spiking neuron. Image adopted from (Floreano and Mattiussi, 2008).

In the current work spiking neurons were used due to their fidelity with regards to the response properties of cortical cells. More specifically, the non-linear membrane potential reset after each spike, the refractory period, and the all-or-none spike afterpotential mechanism (Gerstner and Kistler, 2002) lead to neuronal interactions that have analogies to the operation of real cortical networks. From a computational perspective these three properties give rise to network interactions that are non-linear and semi-chaotic. This creates a biological faithful setting, in which it is interesting to investigate how cognitive mechanisms such as observational learning can unfold.

For the development of the two simulated agents we have adopted the standard form of the Leaky

Integrate and Fire (LIF) neuron model (Stein, 1967) due to the fact that it can encode the three aforementioned properties without requiring an excessive computational overhead. The internal dynamics of each neuron are described by a differential equation (eq. 4.1) that models the fluctuations of the membrane potential variable *u* due to the driving current *I* passing through the neuron:

$$\tau_m \frac{\partial u}{\partial t} = -u(t) + RI(t) \quad (4.1)$$

where R and τ_m are the resistor and membrane potential time constants. In LIF models, spikes are characterized by their firing time $t^{(f)}$, which is the moment that the potential crosses a threshold value *(ϑ)*:

$$t^{(f)}: u\left(t^{(f)}\right) = \vartheta \quad and \quad \left.\frac{du(t)}{dt}\right|_{t\,=\,t(f)} > 0 \quad (4.2)$$

After the emission of a spike, the membrane potential is reset to a constant value $u_r < \vartheta$

$$\lim_{t \to t(f); t > t(f)} u(t) = u_r \quad (4.3)$$

The spike response model used for each presynaptic neuron model is the Dirac (δ) function.

$$S_j(t) = \sum_f \delta\left(t - t^f{}_j\right) \quad (4.4)$$

The spike responses, reduced to points in time through eq. (4.4) are scaled by the weight w_{ij}, of the synapse between the pre-synaptic neuron *j* and the post-synaptic neuron *i*, and summed to construct the input *I(t)* of the post-synaptic cell *i*.

$$I_i(t) = \begin{cases} \sum_j w_{ij} S_j(t) & if\ t - t(f) > ref \\ 0 & if\ t - t(f) \leq ref \end{cases} \quad (4.5)$$

The absolute refractoriness of each neuron, i.e. the time period after the emission of a spike where the input current has no effect on the membrane potential, is modeled by setting the input current to zero for a short time period (ref) after each spike emission.

Due to their rich set of properties, spiking neurons have been extensively used by computational scientists in order to model the cortical properties of the cerebral cortex. One of the most popular neural networks that employs spiking neurons is the Liquid State Machine, which in the current thesis, was employed during the development of the second artificial agent. In the next sections we outline the properties of this network and suggest a method to optimize its computational capacities.

4.3.3 Liquid State Machines

As already mentioned, the most noteworthy property of spiking neurons is their ability to preserve the spatio-temporal properties of their input. This principle has recently been employed in order to construct networks that can perform a large number of computations, known as Liquid State Machines (LSMs). Following their introduction (Maass et al., 2002), LSMs have been used in various pattern classification tasks, including speech recognition (Verstraeten et al., 2005) and movement prediction (Burgsteiner et al., 2007). The notion behind LSMs has also been extended to problem domains outside computational modeling, where researchers use physical mediums for the implementation of the liquid, such as a bucket of water (Fernando and Sojakka, 2003) or real cell assemblies (Dockendorf et al., 2009). In the current thesis, the LSM has been used extensively in order to model the cortical regions of the artificial agent described in chapter 6. For this reason, in the following sections, we carry out an extensive analysis of its classification capability, and suggest a measure that can improve its quality.

An LSM consists of three components: *(i)* the liquid, i.e. a pool M of spiking neurons that accepts input from different sources and outputs a series of spike trains, *(ii)* a filter L that is applied on the output of the liquid in order to create a state matrix S, and *(iii)* one or more memoryless readout maps that are trained to extract information from S (Fig. 4.6). The main conception behind this setup is that the complex dynamics of the input are transformed by the liquid to a high dimensional space, in a way that preserves their recent and past properties. This can be compared to a pool of water with a stone thrown in it. The disturbances that are caused in the liquid could be used by a trained observer to deduce the properties of the motion of the stone before entering the water.

To improve the classification performance of an LSM, one must ensure that for two different input histories, the liquid states produced are significantly different (Legenstein and Maass, 2007). This property, known as *separation*, has recently received increased attention in the literature, due to its close correlation with the performance of LSMs. In (Maass et al., 2002) the separation between two different liquid states is calculated by measuring the Euclidean distance of their state vectors, i.e. the filtered neuron output sampled at one time instance. A similar geometric interpretation has been given by (Goodman and Ventura, 2006), in which the separation of the liquid is measured as the Euclidean distance between the centroids of the states that belong to different classes. In (Dockendorf et al., 2009), the authors use spike train distance metrics instead of Euclidean distance. From the perspective of a classification system, it has been suggested that the rank of the state matrix S can be used to measure the quality of the liquid (Legenstein and Maass, 2007). According to this measure, the larger the number of linear independent variables produced by a liquid state, the better the classification that can be performed by the LSM.

Attempts to improve the performance of an LSM in the literature have shown that it is very difficult to devise a proper measure or structural criterion to optimize the quality of the liquid. For example in (Matser, 2010) the authors have concluded that randomly generated liquids outperform any attempt to structurally modify the LSM. In (Kok, 2007) the authors use both reinforcement learning and genetic algorithms in order to optimize the classification performance of the LSM. Finally in (Norton and Ventura, 2010) the authors use the centroid separation measure in order to drive the synaptic modification of the LSM.

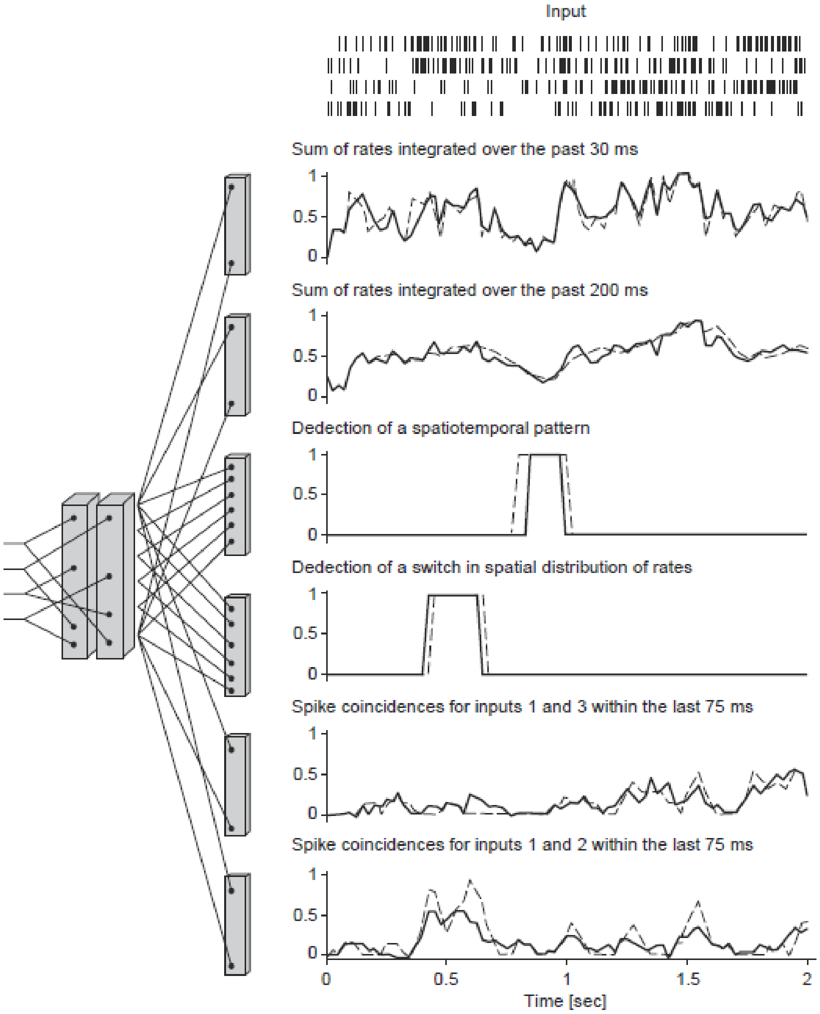


Fig. 4.6. Four readouts reading information from a liquid of a Liquid State Machine. Image adopted from (Arbib, 2003).

In the current section we propose a criterion that measures the separation of the liquid states that correspond to different classes based on the class means and variances (Hourdakis and Trahanias, 2011a; Hourdakis and Trahanias, in press a). The classification performance of an LSM is subsequently improved by employing an evolutionary framework to minimize the introduced criterion. In addition we present experimental results, which attest on the performance and accuracy of the approach, and discuss the benefits of the proposed measure.

Separation Property

Similarly to Support Vector Machines, the LSM acts as a kernel that transforms the low-dimensional space of an input signal to the spatio-temporal space of the liquid. When used for classification, it is important that this transformation yields liquid states that are well separated across different classes.

This separation can be geometrically quantified by employing simple criteria to describe how class vectors scatter throughout the domain space. In the current work, we quantify the separation property by employing two measures. The first requires the means of different classes to be as far away as possible from each other, while the second that the class variances are minimal. The reason that these two measures signify the extent to which different classes are separated is illustrated in Fig. 4.7, for a 2-dimensional, two-class case.

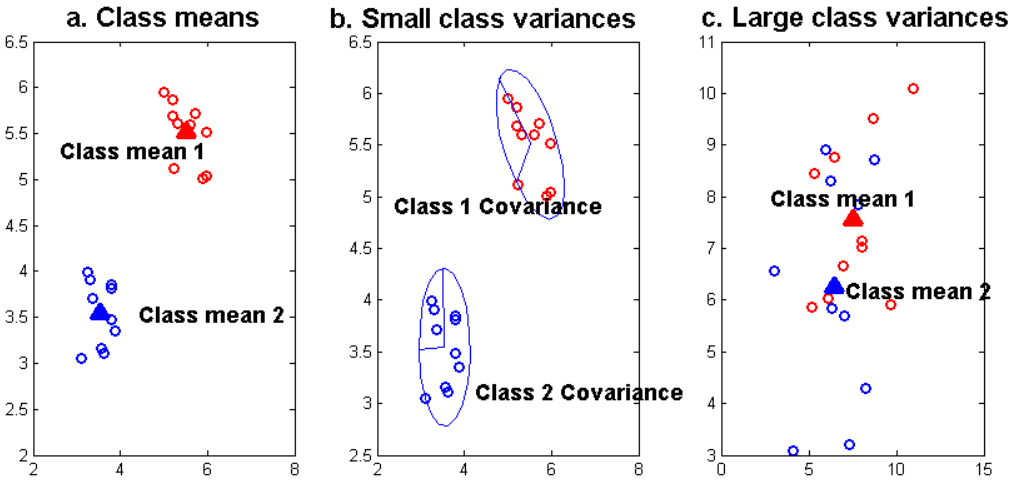


Fig. 4.7. Graphical illustration of the two measures used to quantify the separation of class data. Class 1 (blue circles) is well separated from class 2 (red circles) if (a) the class means are as far away as possible from each other and (b) the class variances are small. (c) An example of how data points from classes with large variance can overlap.

Figure 4.7 shows how, in the 2-dimensional space, the means and variances of different classes can be used to measure the separation of the dataset. The first measure refers to the means of the two classes, and ensures that the baricenters of the data points are geometrically as far away as possible (Fig. 4.7a). The second requires that the class variances are small, so that their respective points will not overlap (Fig. 4.7b). To illustrate this concept, Fig. 4.7c shows how two classes with large variances can overlap, despite them having well separated means.

In the following we describe how the separation property can be implemented computationally, and applied to an LSM in order to measure its classification capacity.

Measure formulation

Based on the separation property, discussed above, in the current section we describe a measure that can quantify the liquid's quality. Accordingly, to evaluate a classification task that refers to ω_i different classes, we define three quantities:

(a) The between-class scatter matrix:

$$S_b = \sum_{i=1}^{M} P_i\,(\mu_i - \mu_0)(\mu_i - \mu_0)^T \quad (4.6)$$

where $\mu_0 = \sum_{\iota=1}^{M} P_i \mu_i$ is the global mean vector for all M classes, P_i is the a priori probability of class ω_i, and μ_i is the mean of the liquid states that correspond to class ω_i. The S_b matrix is a measure of the average distance between the class means and the global mean of the dataset.

(b) The within-class scatter matrix:

$$S_w = \sum_{i=1}^{M} P_i \Sigma_\iota \quad (4.7)$$

where Σ_i is the covariance matrix of the data that belong to class ω_i, and P_i is as in eq. (4.6). Consequently, the sum of the elements of the main diagonal in matrix S_w corresponds to the average variance of all the features in the dataset.

(c) The covariance matrix S_m with respect to the μ_0 global mean:

$$S_m = S_w + S_b \quad (4.8)$$

As eq. (4.8) shows, the S_m matrix can be used as a measure of the sum of variances of the features around the global mean. Consequently, the separation of different classes can be measured by considering a quantity that is proportional to the trace of the S_m matrix and inversely proportional to the trace of the S_w matrix, as described in eq. (4.9):

$$FDR = trace\{S_w^{-1} S_m\} \quad (4.9)$$

which is the generalization of the Fisher Discriminant Ratio (FDR) (Fisher, 1936) to more than two classes. For a one dimensional, two-class problem, it is evident that for equiprobable classes, the matrix S_w is proportional to ${\sigma_1}^2 + {\sigma_2}^2$ while S_b is proportional to $(\mu_1 - \mu_2)^2$, where μ_1, σ_1 and μ_2, σ_2 are the class means and variances. In the next sub-section we will outline how eq. (4.9) can be used to quantify the separation of the liquid states for different classes.

Integrating the FDR measure with the liquid states

To compute the FDR measure (eq. 4.9) one must obtain a representation that contains discrete class vectors for each dataset. For an LSM, this means that the continually changing dynamics of the liquid states must be transformed into a set of discrete values. To accomplish this, in the current paper, we represent all liquid states as a set of discrete geometric locations in a n^{th} dimensional input space, where n is the number of neurons in the liquid. To create this representation, we excite the LSM with a task stimulus, and simulate its neurons for a certain period of time. The discrete spike-trains output in each liquid state are filtered using the exponential function in order to obtain a continuous signal that preserves the intensity of the spike train in the temporal domain:.

$$s_n = \exp(o_n) \quad (4.10)$$

where o_n is the spike train output by neuron n. To apply the FDR measure, we sample the spatio-temporal patterns of the liquid's filtered action potentials with respect to each input, and construct a state matrix S_i for each respective class i, as shown in eq. (4.11):

$$S_i = \begin{bmatrix} s_{11} & \cdots & s_{1j} \\ \vdots & \ddots & \vdots \\ s_{i1} & \cdots & s_{ij} \end{bmatrix} \quad (4.11)$$

where s_{ij} is the filtered output of the j^{th} spiking neuron in the liquid and i is the index of the time window in which the outputs of the neurons are sampled. For each task, we create n matrices S, each corresponding to a different class. The FDR measure can then be calculated by obtaining each class's mean and covariance matrix, and applying eqs. (4.6-4.9).

Graphically, the concept can be illustrated in the following figure (Fig. 4.8). The discrete spike trains that are output during simulation are converted to continuous signals using an exponential filter. The output of this filter is then subsequently sampled with a resolution dt, in order to obtain the data for the separation measure.

In the next section, we describe how the steps of sampling and computing the FDR measure, outlined above, can be integrated in an evolutionary framework that will optimize the performance of an LSM.

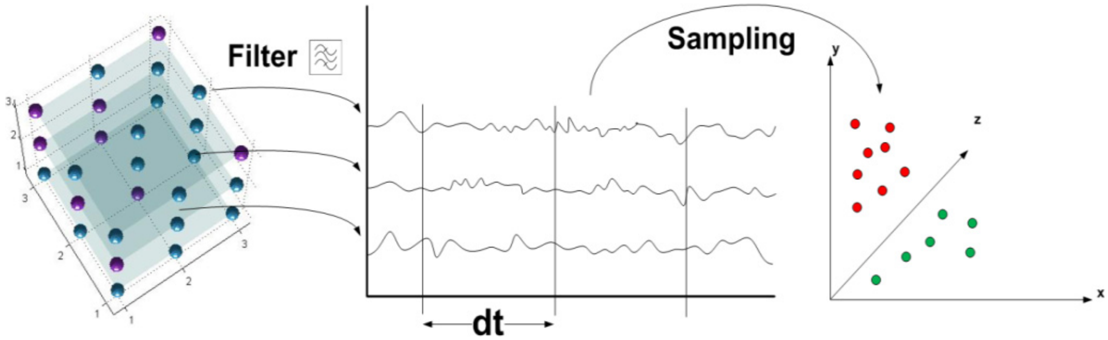


Fig. 4.8. Transforming the output of a Liquid State Machine into a geometrical representation, by filtering the liquid states, and subsequently sampling them every dt steps.

Genetic Algorithm based Liquid Evolution

As mentioned in the introduction, the separation of the liquid is positively correlated with the performance of the LSM (Legenstein and Maass, 2007). Consequently, one can improve the classification performance of the trained readouts by designing a liquid that has a high separation measure. To accomplish this we minimize the FDR (eq. 4.9) of a liquid, using genetic algorithms (GAs). GAs are a stochastic optimization method that can optimize complex functions by exploiting their parameter space (Fogel, 1994). Due to their ability to find good solutions in multi-modal and non-differentiable functions, they have been extensively used to optimize the performance of neural networks. In these cases researchers encode properties such as the architecture, weights or neuronal models of the network and exploit them in order to minimize some objective optimization function (see Yao, 1999 for a review). To evolve the LSM we utilize three types of properties: *(i)* the parameters of the neurons in the liquid, *(ii)* the architecture of the liquid, and *(iii)* the local properties of each architecture. Each of these properties is encoded into a different section of the chromosome that will be employed by the Genetic Algorithm to fine tune the LSM. The effect these parameters have on the liquid performance is discussed in the following.

The first part of the chromosome encodes the firing threshold of all neurons in the liquid and the mean of the Gaussian noise added to the neurons' output on every step. These two parameters control the responsiveness and generalization properties of each neuron. In the first case, if the firing threshold of a neuron is low, then it will require to integrate more spikes before firing a post-synaptic potential, making the liquid less responsive to the perturbations of the low input signals. The second parameter controls the mean of the white noise added to each neuron, which affects the generalization properties of the training.

The second part of the chromosome encodes three different architectures for the LSM. Each architecture specifies a different way for connecting the inputs to the liquid, and the inter-liquid connectivity (Fig. 4.9).

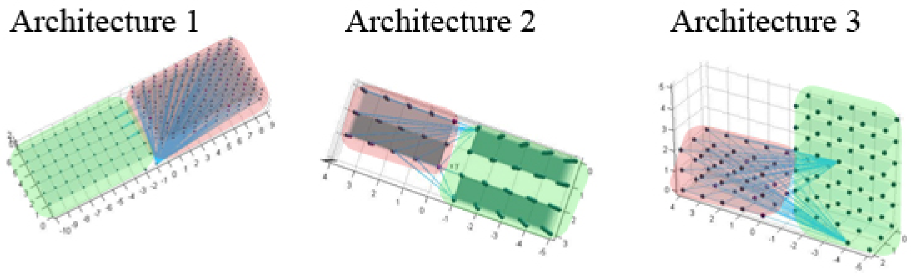


Fig. 4.9. The three architectures used in the genetic algorithm, shown from a different perspective in order to highlight how the components are connected together. In each architecture the liquid component of the LSM is shaded with red, while the inputs with green.

In architecture 1, all input neurons are connected to all neurons within the liquid. Thus all task information is integrated in overlapping liquid locations. This is the original setup suggested in (Maass et al., 2002). In the second architecture the neurons that encode the input from different sources project to different locations in the liquid. In this case, the resulting LSM will produce a state vector whose entries correspond to particular input properties. The third architecture also incorporates the properties of the input, but discriminates it depending on whether they are temporally varying or static throughout the classification task. Temporally varying inputs project to different locations within the liquid, while the constant input signals are propagated to all neurons.

The last part of the chromosome encodes the properties common to all architectures. These include the size of the liquid map and the locality of the connections in the liquid (i.e. the size of the neighborhood that each neuron is allowed to connect to). Because of the exponential descending output of the dynamic synapses of the LSM (Rieke, 1999), the last parameter affects the chaotic dynamics of the liquid, i.e. the period in which a certain input has an effect on the liquid state.

To evolve the chromosomes we use 3 different operators, mutation, crossover and selection. Mutation was implemented by adding a random number drawn from a Gaussian distribution, with zero mean and standard deviation that starts from 1 and decreases linearly until it reaches 0 in the final generation. To perform crossover, the GA selects (with probability 0.5) a bit from each parent chromosome in order to form a child. Selection was implemented using a roulette wheel function.

LSM implementation

To implement the neurons in the liquid we use the Leaky Integrate and Fire (LIF) model (Stein, 1967), because it is a computational convenient way to simulate spike dynamics. However since we aim at optimizing the model through an optimization procedure, in this case we enrich the differential equation of the neuron with additional parameters, in order to allow the GA to exploit them during evolution. In this version of the LIF model, the evolution of the membrane potential is governed by the following equation:

$$\tau_m \frac{dV_m}{dt} = -(V_m - V_{rest}) + R_m * \left(I_{syn}(t) + I_{inject} + I_{noise}\right) \quad (4.12)$$

where V_m is the membrane voltage, $\tau_m = C_m * R_m$ is the membrane time constant, R_m is the membrane resistance, C_m is the resistor capacitance, I_{inject} is a constant current injected to the neuron and I_{noise} a Gaussian random variable with zero mean and a small variance noise. After the emission of a spike, the membrane potential is reset to its resting value V_{rest}. $I_{syn}(t)$ is the incoming current from the presynaptic neurons, and is calculated according to the following equation:

$$I_{syn}(t) = \begin{cases} \sum_j w_{ij} EPSP_j(t) & \text{if } t - t(f) > ref \\ 0 & \text{if } t - t(f) \leq ref \end{cases} \quad (4.13)$$

The absolute refractoriness of each neuron, i.e. the time period after the emission of a spike where the input current has no effect on the membrane potential, is modeled by setting the input current to zero for a short time period (ref) after a spike emission at $t(f)$. $EPSP_j$ is the output of the j^th pre-synaptic neuron, t is the current simulation time, while $\mathrm{w_{ij}}$ is the weight connecting the presynaptic neuron i and the postsynaptic neuron j.

Neurons within the liquid are connected with the dynamic synapse model suggested by (Markram et al., 1998). In this model, a synapse's n postsynaptic potential ($\mathrm{EPSP_n}$) changes dynamically due to the arrival of new spikes. It is governed by the following equations:

$$\mathrm{EPSP_n} = \mathrm{K} * \mathrm{R_n} * \mathrm{u_n} \quad (4.14)$$

$$\mathrm{u_{n+1}} = \mathrm{u_n} \exp\left(-\frac{\Delta \mathrm{t}}{\tau_{\mathrm{facil}}}\right) + \mathrm{U}\left(1 - \mathrm{u_n} \exp\left(-\frac{\Delta \mathrm{t}}{\tau_{\mathrm{facil}}}\right)\right) \quad (4.15)$$

$$\mathrm{R_{n+1}} = \mathrm{R_n}(1 - \mathrm{u_{n+1}}) \exp\left(-\frac{\Delta \mathrm{t}}{\tau_{\mathrm{rec}}}\right) + 1 - \exp\left(-\frac{\Delta \mathrm{t}}{\tau_{\mathrm{rec}}}\right) \quad (4.16)$$

The maximum output of the synapse is governed by the absolute synaptic efficacy K. The change of the efficacy is determined using the variables $\mathrm{u_n}$ and $\mathrm{R_n}$, which are calculated using eqs. (4.15) and (4.16), respectively. τ_{facil} and τ_{rec} are constant parameters, experimentally specified. $\mathrm{u_n}$ defines the utilization of the synaptic efficacy which decays exponentially based on the τ_{facil} parameter to its resting value U. $\mathrm{R_n}$ is the fraction of available synaptic efficacy and defines the strength of the $\mathrm{EPSP_n}$ at a given spike. It reduces due to the arrival of new spikes and recovers exponentially according to the τ_{rec} parameter. At t=0, the following initializations occur: $\mathrm{u_1} = \mathrm{U}$ and $\mathrm{R_1} = 1$. Δt is the time difference between the n$^{\text{th}}$ and (n+1)$^{\text{th}}$ spike.

At the initialization of a simulation all neurons are placed in a 3-dimensional grid and are assigned a triplet of $\mathrm{x, y, z}$ coordinates. In the liquid, the probability that two neurons are connected is governed by the following equation:

$$\mathrm{p_c(a, b)} = \mathrm{C(a, b)} * \mathrm{e}^{\frac{\mathrm{D(a,b)}^2}{\lambda^2}} \quad (4.17)$$

Equation (4.17) defines the connection probability of two neurons being connected, according to their distance in this grid. a denotes the presynaptic neuron and b the postsynaptic neuron. The constant $C(a, b)$ takes different values according to the excitation status of the pre- and post-synaptic neurons (i.e. whether the neurons are inhibitory (I) or excitatory (E)). These are set to 0.2 for EE, 0.3 for EI, 0.4 for II and 0.2 for IE. $\mathrm{D(a, b)}$ is the Euclidean distance between the two neurons, while λ scales the average length of each connection.

Experimental Results

In the current section we evaluate the performance of the FDR measure and the GA optimization framework on a number of different classification tasks. For this reason, we focus on two issues: *(i)* the ability of the proposed measure to predict the quality of the liquid in an LSM, and *(ii)* whether the optimization framework can reduce the error of the readouts by minimizing the FDR.

Comparison with popular measures in the literature

For the first classification task, we compare the performance of the two most popular measures in the literature, namely the centroids and rank measures, against the FDR. For this reason we use an LSM with one linear regression readout to classify whether the rate of the input is above a particular value, in this case five Hertz. Input is encoded as a random Poisson rate and applied for 1000ms to a pool of LIF spiking neurons. The liquid used for this purpose consists of a pool of 125 spiking neurons, arranged in a 3 dimensional grid. States are sampled every 10ms for 1 second and input to 4 readout units.

The FDR, Rank and Centroids measures were *(a)* calculated for the two state vectors S_1 and S_2 from eq. (4.10) that correspond to classes ω_1 and ω_2, and *(b)* compared against the performance of a linear regression readout for 16 different simulations (Fig. 4.10).

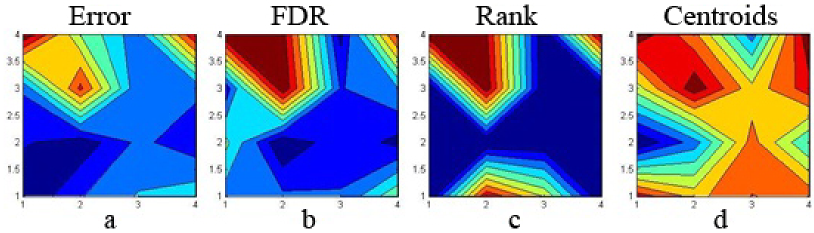


Fig. 4.10. Evaluation of the FDR, Rank and Centroids measures against their ability to predict the performance of the linear readout map of an LSM. In each subplot the x,y axes correspond to different configurations for an LSM. The color in each x,y entry corresponds to the value of the error (for subplot a) or the negative value of the measure (for subplots b,c,d).

A good measure should be positively correlated with the error of any trained readout that is used to extract information from the liquid. As Fig. 4.10 shows there is a clear correlation between the value of the error of a readout map and the value of the FDR measure (for both cases high values close to 1 are colored with red shades, while low values close to 0 are colored with blue). Furthermore, the results presented in Fig. 4.10 show that the proposed measure outperforms the Centroids measure and, at the same time, performs better than the Rank measure. By comparing Fig. 4.10a and Fig. 4.10b, it is evident that the FDR measure can predict with satisfying accuracy the performance of the linear regression readout and, therefore, the separation of the liquid in the LSM (readout error/FDR correlation was 0.86). Due to space limitations we do not provide the corresponding contour plots for the other three readouts, although we note that the results were similar to the ones presented in Fig. 4.10.

Measure evaluation

In the current section we consider three additional classification tasks, in order to evaluate whether the FDR measure can predict the performance of an LSM while solving them. Each task incorporates a different method for encoding the input. This is important since diverse input encodings can have a different effect on the liquid dynamics. Population codes (Rieke, 1999) provide a consistent representation of the input by using distributed and partially overlapping neuron groups in order to encode the values of a variable. In contrast, rate codes (Rieke, 1999) produce a higher homogeneity when used as input because they employ the same neuron to represent different input values.

Consequently in the three tasks discussed below, rate codes are used to encode the input in two cases, whereas population codes are used in the third case. The liquid in all the aforementioned classification tasks follows the initialization and topology settings discussed in (Maass et al., 2002). To evaluate the measure on different classification methods we employ four different readouts: the first readout is implemented with a multi-layer perceptron using the backpropagation rule to train the weights (Rumelhart et al., 1986). The second readout implements linear regression (Duda et al., 2000). The third readout implements a classifier that uses least squares to find the regression coefficients (Duda et al., 2000), while the fourth the p-Delta rule on a parallel perceptron layer (Auer et al., 2008).

Classifying different behaviors

The first task requires the LSM to classify two different motions of an object, based on the projection of its image on a 9x9 grid of receptive field neurons. The output of the retina field is encoded as a group of 81 neurons, each one corresponding to a different cell. These neurons fire random Poisson spikes of 30Hz when their corresponding cell in the retina field is occupied, and at a rate of 5Hz otherwise. This output is then projected to a liquid with 63 neurons, where we record the post-synaptic potentials of the neurons for 3 sec (3000ms).

The LSM is used to classify two different motions of an object, when it is projected on a 9x9 grid of receptive field neurons. The following figure illustrates a sample motion of 5sec duration.

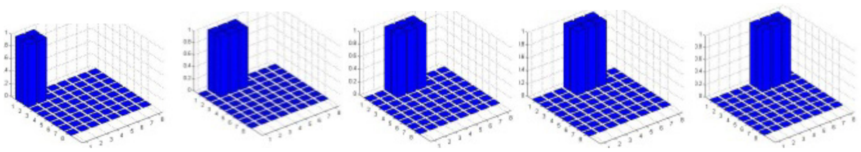


Fig. 4.11. The activations of the neurons in the retina field as the object moves through different locations. Neuron activations are spread through a neighborhood of size 1, i.e. for each location, the neighboring neurons are also activated.

Information from the liquid response is classified using four readouts: (i) Linear regression, (ii) Feedforward neural network, (iii) Linear classification and (iv) p-Delta rule. The liquid must learn to classify whether the movement on the retina belongs to either one of the two behaviors for 9 different simulations. The error is calculated by subtracting the readout value from the actual behavior being performed for each step of the simulation, and normalized to 1. Due to space limitations, results are not presented for this task in the form of contour plots but rather as graphs of the errors of the readouts against the FDR (Fig. 4.14a).

As Fig. 4.14a illustrates, FDR follows quite closely the corresponding error in all cases. The correlation between the FDR measure and the readout error was in all cases above 0.8, indicating a close relationship between the two.

Classifying different objects types based on their shape

For the second task we use an LSM that must classify the type of three different objects, a circle, a square and a hexagon, based on their images. To encode the input we first sharpen each image using a Laplacian filter and consequently convolve it with 4 different Gabor filters with orientations π, $\frac{\pi}{2}$, 2π and $-\frac{\pi}{2}$ respectively (Fig. 4.12).

The four convolved images from the input are projected into four neuronal grids of 25 neurons, where each neuron corresponds to a different location in the Gabor output. Information from the liquid response is classified by the above mentioned four readouts for 9 different simulations.

$$\frac{1}{(a+1)}\begin{bmatrix} -a & a-1 & -a \\ a-1 & a+5 & a-1 \\ -a & a-1 & -a \end{bmatrix}$$

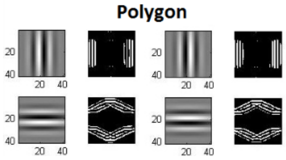


Fig. 4.12. The Laplacian filter used to convolve each image (left) and the result of the convolution of the 4 Gabor filters with the polygon image (right). In each row on the right image we show two pairs of the filter in different directions (left subplot for each pair) and the result of the convolution (right subplot for each pair).

As the four plots in Fig. 4.14b show, the FDR measure (blue line) is able to predict with high accuracy the classification error for each one of the four readouts used (read line in each plot).

Classifying the location of an object on a retina field

For the third task, we use an LSM that must classify whether an object is on a certain location upon a grid. To encode the spatial representation of the environment we use 25 Poisson neurons that become active only when the object's position is upon their respective location in the grid. Neurons fire with an intensity of 50Hz if they are located at the center of the object's position, while at 20Hz if they are placed in one of the neighboring cells. Figure 4.13 illustrates the representation and encoding of the stimulus used for the LSM, for two different positions.

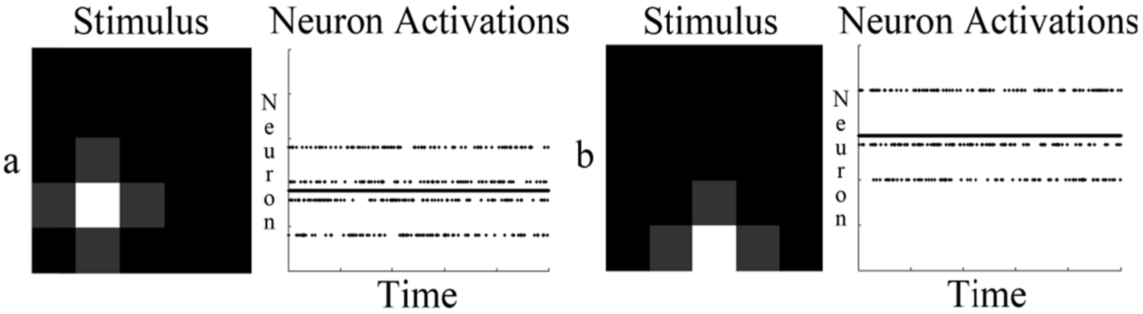


Fig. 4.13. The representation of the retina field of the input and the encoded stimulus representation of this field. The plot demonstrates two different positions. a. Retina field activations for position 15 (left) and spike trains generated for the LSM (right). b. Retina field activations for position 9 (left) and spike trains generated for the LSM (right).

The output of the retina field is projected onto an LSM, which is simulated for 1 sec (1000ms). The liquid states, after being filtered and sampled, are used to calculate the FDR measure for each different position, i.e. the classification contains 25 classes, each corresponding to a different location in the grid. In Fig. 4.14c we illustrate how the FDR measure was able to predict the performance of the LSM for the current task, for all four readouts. The results presented above for all three tasks, indicate that the FDR measure can describe the quality of the liquid over a broad range of tasks and input encodings. Having established an accurate measure of the quality of performance of the LSM, in the following section we examine whether a GA framework can optimize the performance of the classification performed by the liquid.

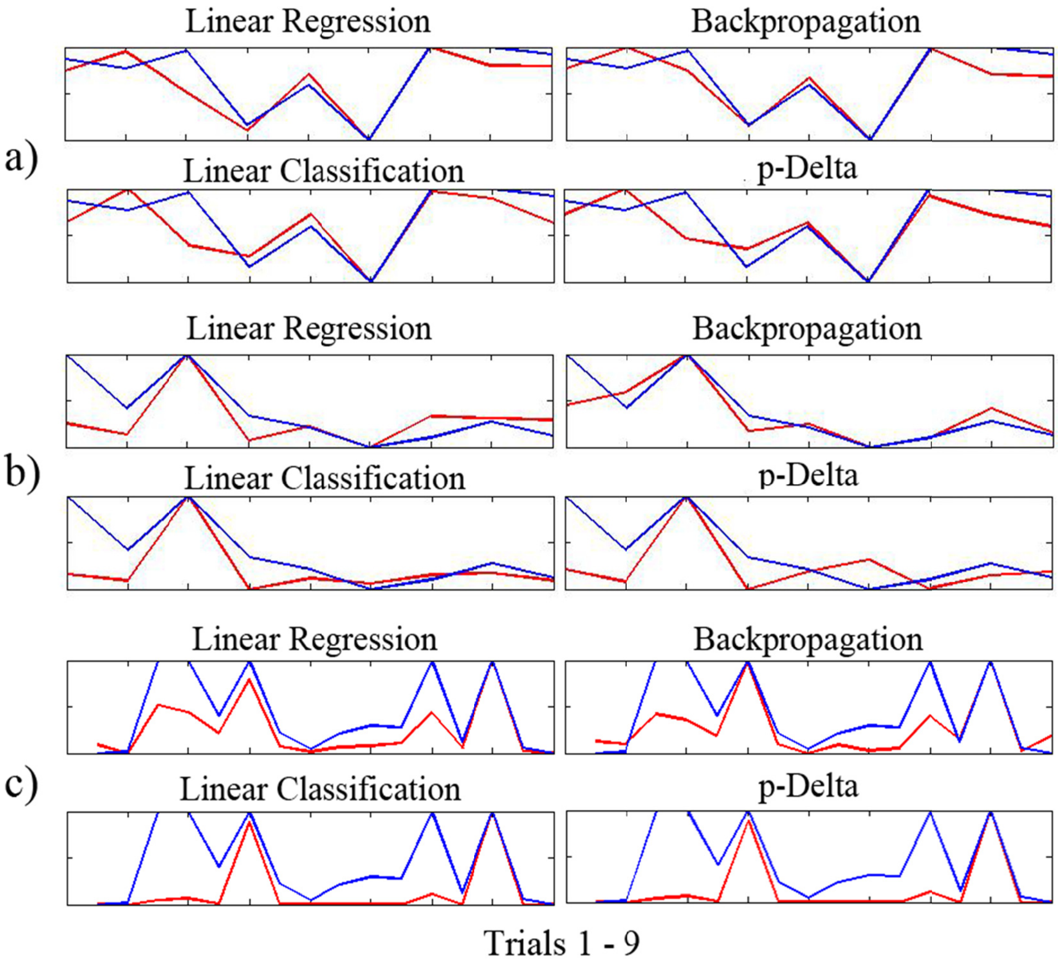


Fig. 4.14. Graphs of the FDR measure (blue line) against the readout error (red line) of linear regression, backpropagation, linear classification and p-Delta methods. The x-axis corresponds to the 9 different trials of the simulation, while the y-axis shows the output of the corresponding error and measure.

Liquid optimization

In the current section we evaluate the extent to which the aforementioned GA framework can optimize the performance of an LSM. For this reason, we consider an additional classification task that requires the integration of temporal information in the liquid states. More specifically, we consider a binary classification task in which an LSM must classify whether the end point of a moving planar robotic arm is closer (or not) than a predefined distance to a given target location. The difficulty of the task lies in the fact that the time-varying control model of the arm must be combined with the static signal of the end point location and produce discrete liquid states, even in cases where the input dynamics are not so different.

Input in this case consists of two different channels. The first encodes the spatial location of an end point position in (x, y) space coordinates and the second the inverse kinematics of different arm trajectories. Input joint positions are generated by creating different trajectories using a two-link planar arm based on one start position (Fig. 4.15a), three speed profiles (Fig. 4.15b) and five random ending positions for the train and test sets (Figs. 4.15c,d). The training set consists of the trajectories between the initial position (Fig. 4.15a) and a random end position (Fig. 4.15c). The test set is generated using a different set of ending positions (Fig. 4.15d).

To determine the trajectory between a starting and ending position, a random speed profile is chosen from the templates in Fig. 4.15b. The joint configurations of the robot across the pathway of a trajectory are obtained using an iterative solution to the inverse kinematics problem, based on the pseudo-inverse of the robot's Jacobian.

To encode the target position we use a population code with 10 neurons for each dimension (i.e. the x, y coordinates). Thus for the two dimensional space 20 input neurons are used. The simulated robotic arm that is employed in the experiments consists of 2 joints, namely elbow and shoulder, whose values are also encoded using population codes.

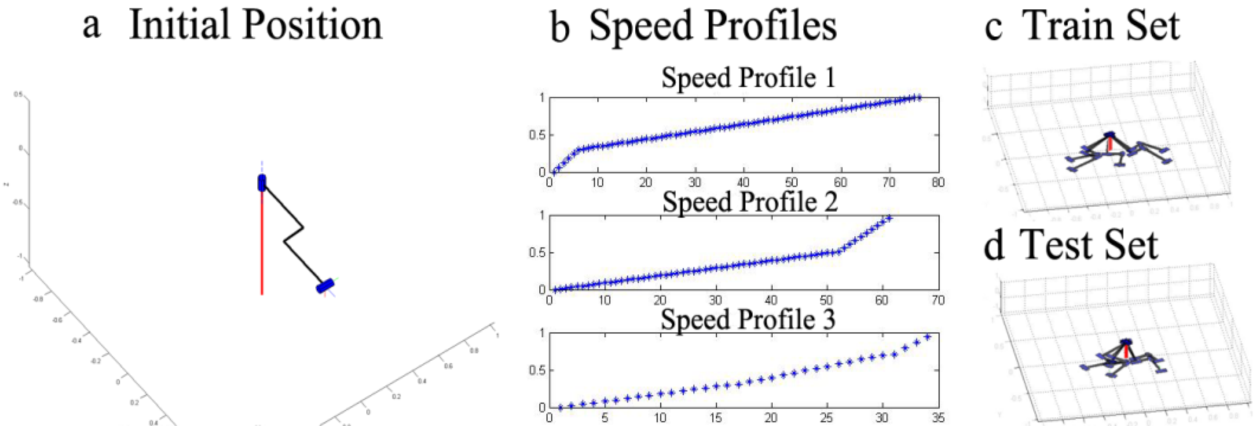


Fig. 4.15. a. The initial position of the robot's arm. b. The three speed profiles used to generate random movements. c. The five different configurations of the arm of the robot for the five ending positions of the train set. d. The five different configurations of the arm for the test set.

The classification task we consider requires the LSM to predict whether in the next location, the end point of the robot's arm will be closer than a predefined distance to the object in x, y coordinates. To classify a given location correctly, the LSM must make a prediction on the speed of the arm at any given time. Hence, the liquid state must integrate information about the location of the robot's end-point effector position in previous time steps. To generate the different liquid states we conducted 100 simulations for the train set kinematics and 30 simulations for the test set kinematics. To learn the classification task, these liquid states were input to 4 readout units, namely a feedforward neural network, a parallel perceptron layer, a linear regression and a linear classification readout.

Even though the results reported here regard the optimal individual produced by the genetic algorithm, it should be noted that similar results were obtained for all chromosomes in the last generation. The GA was run for 18 generations in order to minimize the FDR criterion (Fig. 4.16). As the four rightmost plots in Fig. 4.16 show, while the genetic algorithm was used to minimize the FDR measure, it also reduced the error on all four readout maps.

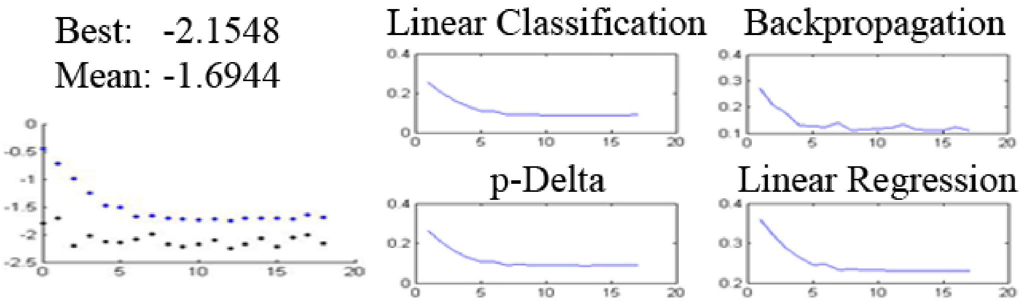


Fig. 4.16. The results from the evolution of the LSM for 18 generations of the 100 individuals. The left plot shows the best and mean fitness (y-axis) of the population on each generation (x-axis). The right set of four plots shows the average error of the four readouts across the generations (y-axis) of the 100 individuals (x-axis) in the final population.

The subplots in Fig. 4.16 demonstrate how the GA was able to optimize the performance of the LSM by reducing the classification error of the readouts. The error was reduced from 0.3 to 0.1 in all four readouts maps (Fig. 4.16, right four subplots), simply by optimizing the FDR measure on the liquid (Fig. 4.16, left subplot) during the evolutionary process.

The improvement in the liquid performance is also evident when we examine the output of the four classifiers in the optimal individual produced by the GA. As Fig. 4.17 shows, all the readouts are able to classify different movements with a high degree of accuracy. In the first example (plot a), the robot's arm never reaches the target location in a distance closer than required. In the second (plot b), it approximates the end location in the final 10 simulation steps. In both plots, we show the stimulus input to the liquid (top graph of plots a,b), the output of the four readouts (blue lines in bottom four graphs of plots a,b) and the target for each readout (red lines in bottom four graphs of plots a,b). Each graph is labeled with the corresponding readout map. The x-axis represents the simulation time in 100ms intervals for all graphs.

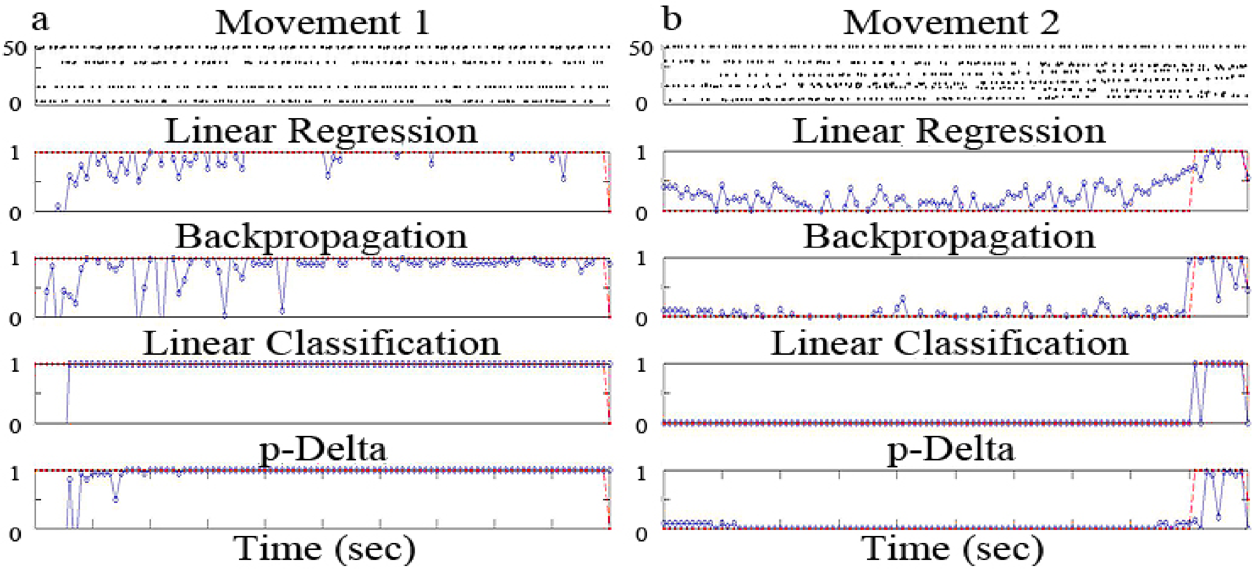


Fig. 4.17. The output of the four trained readouts for two sample movements (four bottom subplots) and the input projected to the liquid (top subplot).

GA framework evolved LSM parameters

In the current section we present the statistics of the genetic evolution with respect to the chromosome parameters for the 30 most optimal individuals in the final generation. These are important since they can describe the strategies employed by the GA to solve the classification task. We also measure the effect that each parameter has on the liquid performance, by calculating its correlation with the error of the linear regression readout.

As Table 4.1 shows, the architecture of a liquid has a significant effect on the performance of the readouts, since its correlation with the error is very high. In the current task, the first architecture produced fitter individuals than architectures 2 and 3. The locality of the connections (λ parameter) and the noise added to each neuron (I_{noise}) are also positively correlated. The medium value of 3 for the λ parameter in the optimal chromosomes shows that local synapses are better suited for connecting the neurons in a liquid. Finally, the threshold of the neurons and the size of the liquid are negatively correlated with the liquid's performance. The latter result has also been pointed out by (Maass et al., 2002).

Having established a concrete modeling framework, in the next two chapters we describe the development of two computational agents that were based on the LIF spiking neuron model. The computational agent described in chapter 5, employs the spike neuron to develop various representations through different plasticity rules. In contrast, the agent we describe in chapter 6 employs the LSM, in order to form a spatio-temporal pattern of spike activations.

Table 4.1. The variables evolved by the GA. For each variable we list the average value in all the chromosomes from the best population, its correlation with the error of the linear regression readout throughout the genetic evolution and the range of permitted values used by the GA.

Parameter	Average value on the best population	Correlation with error	Range used in the GA
[X, Y, Z]	[5, 6, 5]	[-0.18,-0.23, -0.32]	[0..6] for all X, Y, Z
λ	3	0.188	[0..8]
V_{thresh}	0.02	-0.21	[5e-3..5e-2]
I_{noise}	0.0005	0.3	[5e-8..5e-7]
Architecture	1	0.633	[1..3]

Chapter 5

Observational learning based on the cortical underpinnings of Macaque primates

In the current chapter we describe the design and implementation details of the computational model that was developed in order to investigate how the overlapping activations during execution and observation facilitate observational learning. The model is inspired by the neurophysiological data that pertain to action observation in Macaques, and were reviewed in chapter 3. To integrate them within a computational context we combine the pathway methodology and the biologically inspired neural networks that were discussed in chapter 4.

In the following sections we summarize the goals set for the model (section 5.1), and the modeling approach that will be used in order to accomplish them (section 5.2). We then continue to describe the development and implementation details of the computational agent (sections 5.3-5.6), along with an extensive evaluation of its ability to perform observational learning, in a manner similar to its biological counterparts (section 5.7).

5.1 Problem Statement

According to the imaging experiments in (Raos et al., 2004; Raos et al., 2007), Macaques activate additional neural regions during observation, apart from area F5, including cortical areas that are associated with proprioception and motor control. Even though monkeys are not capable of imitation, however they can recognize the actions of other agents and posses the ability to associate known motor patterns to unknown stimulus incentives from the environment (Subiaul et al., 2004). The extended pathway of activations facilitates these abilities by using a broader circuit of brain regions during observation, that extends beyond area F5 and the mirror neurons. Considering the cognitive role of those regions during the execution of a behavior (see section 3.2.2), in the following sections we investigate how a computational agent can recognize actions and associate behaviors from its repertoire of motor actions to novel objects from the environment only by observation.

To make use of the available neurophysiological data, which in monkeys can reach up to the activation of single cells, we have focused on developing the model's representations at the neuronal level. This enabled us to evaluate important assumptions in respect to observational learning which, inline with the three theoretical questions discussed in the introduction (see section 1.4), are summarized below:

1. *What types of representations participate in the formation of the motor image during observation, and what is the contribution of the motor control system in this process?*
   This issue is derived from theoretical question A and pertains to the properties of the mental image that is formed by an observer, and how it can be developed using the motor control system of the agent.

2. *How can the computational agent exhibit behavioral skills similar to the ones found in Macaques, using the pathway of overlapping activations?*
   The second issue that we investigate is based on the theoretical question B, and focuses on how the computational agent can exhibit observational learning, in a manner that is compatible with the cognitive imitation capacities found in Macaques.

3. *How are the action recognition abilities of the Macaque monkeys supported by the activated regions during observation?*
   The final issue that we focus on regards the neurophysiological basis of the system that supports the imitation skills in Macaques, and pertains to theoretical question C. More specifically, based on the high resolution level of the model, it focuses on the types of sensorimotor representations that can be employed by a computational agent, in order to compensate, through mental simulation, for the immobility of its embodiment during observation.

5.2 Modeling approach

To develop the model it is first important to identify the cognitive functions that are performed in the activated regions. In line with the discussion in chapter 3, the regions that become active in Macaques during observation are responsible for: *(i)* executing motor behaviors, *(ii)* maintaining the sensorimotor representations of the movement and *(iii)* handling visual perception.

To implement these functions computationally we need to identify the effect that the model's inputs have to each neural network. To accomplish this we identify certain *pathways* within the model, i.e. streams of network interactions that process different types of information. We then continue to discuss the nature of the representations that can be implemented at the single and population neuron levels, as well as the underpinnings of the input and synaptic learning models that will be used to develop these representations in the computational agent.

5.2.1 Identifying the Model Pathways

In the current section we derive the cognitive functions that will be performed by the model, and identify the regions that will be used to implement them. To accomplish this we employ the pathway modeling methodology, which was described in chapter 4, in order to group the functionality of the regions that become active during observation and execution in Macaques, into three respective functions (Fig. 5.1): *(i)* object perception, *(ii)* proprioceptive association and *(iii)* motor control.

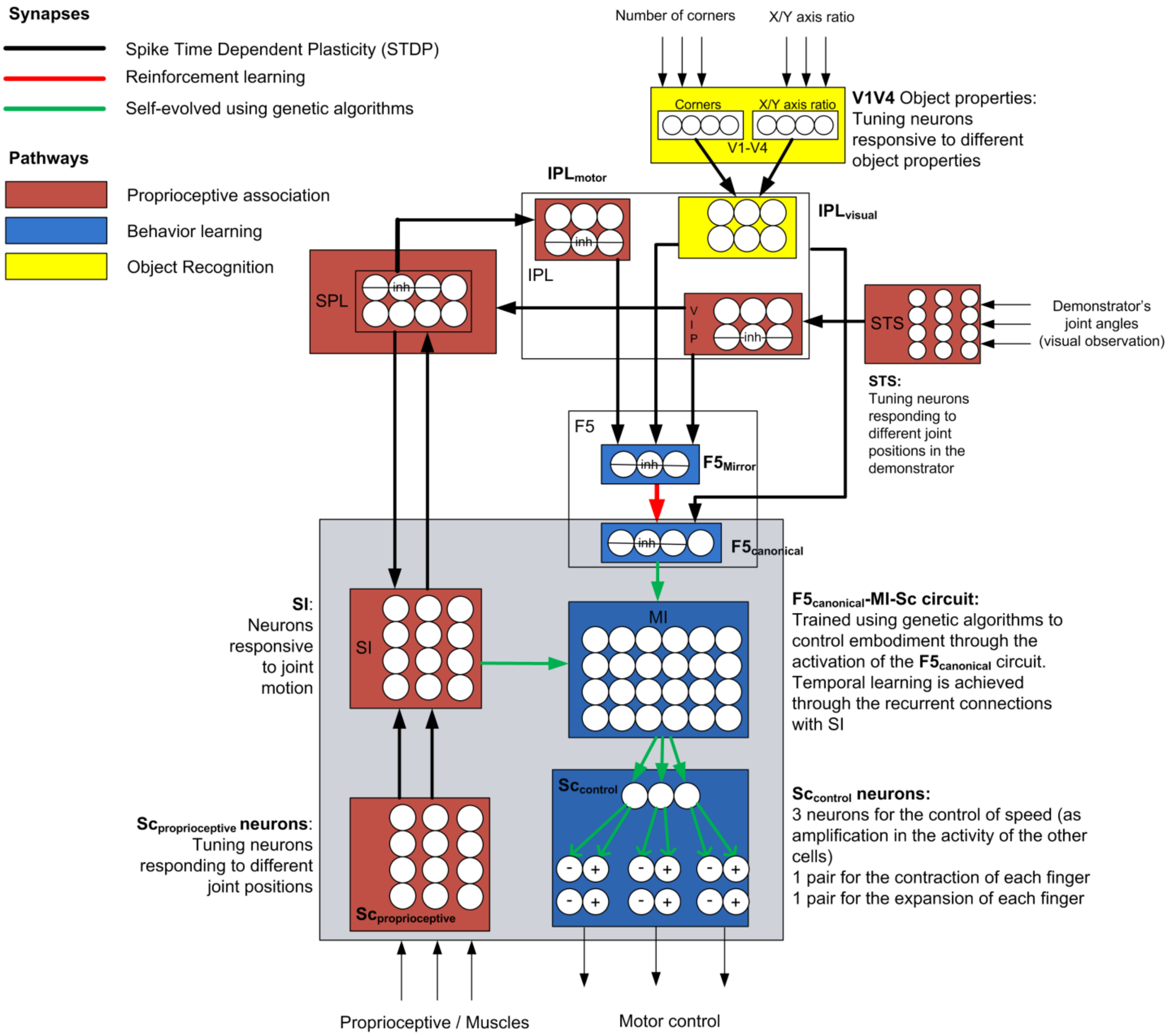

Synapses
Spike Time Dependent Plasticity (STDP)
Reinforcement learning
Self-evolved using genetic algorithms
Pathways
Proprioceptive association
Behavior learning
Object Recognition
Number of corners
X/Y axis ratio
Corners
X/Y axis ratio
V1-V4
V1V4 Object properties: Tuning neurons responsive to different object properties
IPLmotor
inh
IPLvisual
IPL
SPL
inh
V I P
inh
STS
Demonstrator's joint angles (visual observation)
STS: Tuning neurons responding to different joint positions in the demonstrator
F5
inh
F5Mirror
inh
F5canonical
SI: Neurons responsive to joint motion
SI
MI
F5canonical-MI-Sc circuit: Trained using genetic algorithms to control embodiment through the activation of the F5canonical circuit. Temporal learning is achieved through the recurrent connections with SI
Sccontrol
Sccontrol neurons: 3 neurons for the control of speed (as amplification in the activity of the other cells)
1 pair for the contraction of each finger
1 pair for the expansion of each finger
Scproprioceptive neurons: Tuning neurons responding to different joint positions
Scproprioceptive
Proprioceptive / Muscles
Motor control

Fig. 5.1. Layout of the proposed model. The three pathways are marked with different colors; object recognition: yellow; proprioceptive association: red; behavior learning: blue. Different types of synapses are also marked with different colors: STDP: black, reinforcement: red, GA: green. The lines crossing the neurons in some networks (e.g. SPL) indicate the existence of lateral inhibitory connections in the respective networks.

Cortically, *object perception* is facilitated by the regions of the occipital lobe, where visual incentives are integrated together into forming a coherent representation of the incoming percepts. The outcome of these projections is evident in the activity of IPL, where the dorsal pathway ends, which encodes a higher-order representation of the perceived objects. Consequently, the first pathway we identify for the model includes regions V1V4 as well as regions in IPL, and will be responsible for developing a representation of the perceived object.

The second function that is activated during observation is *proprioceptive association*. To facilitate this function the primate must combine the visual incentives from the demonstrator's motion, with its own sensorimotor codes of the movement. Consequently, this pathway will include regions that pertain to the visual perception of body parts (STS), regions that encode the proprioception of the agent (Sc, SI) as well as higher integration areas that are responsible for associating the motor and visual incentives into a common neural code (SPL, IPL).

The final pathway that we focus regards motor control, and employs information from the other two pathways in order to learn and generate a motor behavior. It involves regions in the premotor, parietal (dorsal and ventral) and primary motor cortex (MI), and is used to implement the association between an object representation and a respective behavior, using the connections between the premotor cortex and IPL.

Based on these three pathways, the final goal of model development is to integrate their functions together in order to produce a working model of cognitive imitation, able to exhibit the corresponding imitation capacities found in Macaques (Subiaul et al., 2004). The connectivity between different neural networks is derived in accordance to the connections between the respective cortical areas in the brain (Kandel et al., 2000) and is depicted in Fig. 5.1.

Figure 5.1 presents an outline of the computational model, along with the respective connections between the regions. The object recognition pathway is assigned the role of forming the neural representations that encode the objects presented to the agent. This is accomplished in the $V1V4_{corners}$, $V1V4_{XYaxisRatio}$ and IPL_{visual} networks. The second pathway, proprioceptive association, is responsible for forming the neural representations of the action of the agent, as well as building the correspondence between the actions of the observer and demonstrator agents. It originates from $Sc_{proprioceptive}$ region, and involves the IPL_{motor}, STS, SI, SPL and VIP networks. The last three regions are part of the circuit that performs action correspondence. Finally the third, motor control, pathway involves regions $Sc_{control}$, MI and F5 and will be employed in order to facilitate motor control.

Highlights of the model

The model presented in Fig. 5.1 consists, to the best of our knowledge, the first computational implementation of an artificial agent that uses its motor control system in order to learn during observation. This is accomplished by employing the representations derived from the motor image of the agent during execution, in order to compensate for the spinal cord immobility (Stamos et al., 2010) during observation. To our knowledge, this is the first attempt to model the process of observational learning, i.e expand an artificial agent's motor control system without employing its embodiment.

5.2 Modeling approach

Even though the model is designed in large scale, it considers in detail the neuronal properties of the cortical regions that become active during execution and observation in Macaques. These types of high resolution models are currently missing from the motor control literature, since researchers either consider the neuronal properties at small scale models, or resolve to making several simplification assumptions when designing large-scale ones. In our case, the high-level of detail was facilitated by two important factors: *(i)* the use of the computational pathways methodology, which enabled us to identify how the cortical representations of each region can be coupled together into realizing the model's higher order functions and *(ii)* the high level biologically inspired models of neurons (LIF model) and synapses (STDP, Bienerstock-Monroe) that were used to implement each cortical region. As we will discuss in the following sections, this allowed the model to exhibit several properties that are consistent with the neurophysiological model of action observation/execution in Macaques, including: *(i)* similar regional activations during observation and execution (section 5.7.2) and *(ii)* evidence as to how the overlapping mechanism can facilitate learning during observation (section 5.7.3).

Another important property of the model is that it makes a clear distinction between canonical vs mirror neurons in the Ventral Premotor (F5) region. As discussed in chapter 2, this property is very important in order to discriminate between the neurons that are responsible for action recognition, and the neurons used for encoding the agent's primitive behaviors. As a result, the computational model was able to make an important distinction between the processes of action recognition, and action execution, and employ them accordingly during observation. More importantly, due to the aforementioned fact, the computational neurons that were used to model the mirror neurons in the F5 region exhibited intransitive and associative properties, similarly to their cortical counterparts.

In addition, the results that we present in this chapter show several computationally attractive properties that were exhibited by the computational agent. These include the capacity to (i) learn new behaviors only by imitation, (ii) associate known behaviors to unknown stimulus from the environment only by observation, (iii) preserve previous learned behaviors after a learning cycle and (iv) generalize knowledge to similar objects.

Finally, due to the high resolution that was used, the model allowed us to derive important assumptions regarding to: (i) the reasons for the lower activation in some regions of the cerebral cortex of Macaques, (iI) the type of observation/recognition that can be facilitated by the overlapping pathways and (iii) the role of some regions into performing embodiment correspondence.

The development and implementation of these computational properties is discussed in the remaining sections of this chapter. More specifically we describe: *(i)* the types of neural representations that can be developed based on the biologically inspired neuron model that was used (section 5.2.2), *(ii)* the plasticity and synaptic models employed to develop the agent (sections 5.2.3), *(iii)* the encoding of the visual and proprioceptive input (section 5.3.1), *(iv)* the information processing carried out by each neural network (section 5.3), and *(v)* the setup of the circuitry that is used to produce the output signals for the motor control of the agent (section 5.4).

5.2.2 The nature of the model's neural code

As already mentioned, the regions in the computational model are implemented using networks of spiking neurons. Each network is designed in order to perform a transformation of the input signal into a corresponding symbolic code. Such representations can be developed by manipulating the patterns of neurons and synapses in a network, in order to explicitly design it to exhibit the appropriate neuronal activity.

The building block for developing a symbolic representation is the output of the spiking neuron model that was described in chapter 4. Even though spiking neurons communicate through a single temporal event, the spike after-potential, they can exhibit a variety of voltage dependent behaviors. The most important of these are: post-synaptic potential (PSP) integration, paired-pulse facilitation and depression, rebound facilitation, bursting and slow integration of PSPs (see Zucker, 1989 for a review). These simple properties give rise to complex representational phenomena at both the single neuron and population levels (Durstewitz and Seamans, 2000; Amit and Brunel, 1997), providing different ways for encoding an input stimulus. These are discussed in the two sections below, while in section 5.3 we describe how such neural codes are employed by the computational agent in order to exhibit the appropriate behaviors.

Representations at the single neuron level

The main benefit of spiking neurons over other formal neuron models is that spikes are point processes characterized by their timing. This property adds a temporal dimension to the neurons' output and facilitates very powerful forms of information processing (Maass, 1997). Consequently, spiking neurons can use diverse forms to communicate their message, which in practice means that a stimulus can be encoded by a neuron with different code conventions.

From a computational perspective, the nature of a neuron's code is very important. Two neurons can encode the same stimulus in a different way, thereby creating different representations for the same event. Since plasticity is dependent on the output of the neurons at both ends of a synapse, the way that a neuron chooses to communicate its message has a direct effect on the learning abilities of the network (Fig. 5.2).

The most common way to treat a spike is to consider it as a momentary temporal event with no spatial dimension. This type of neural code, known as *temporal*, has been observed in various areas of the cortex (Mainen and Sejnowski, 1995), e.g. in the auditory system it is used to perform echolocation (Knudsen and Konishi, 1979). Temporal codes have redefined traditional views about cortical representations, by showing that synchronization of neuronal firings across different regions can bind features together. The most common way to simplify the processing in a temporal code is to average it over a certain period of time. This code convention is known as the *rate code*. Rate codes have been observed in stretch receptor neurons, where the force that is applied in the muscle is correlated with the firing rate of the neuron (Adrian, 1926). Some however argue that the cerebral cortex does not have the capacity to evaluate more than one spike at a given time, and therefore it is unable to process rate codes properly (Thorpe et al., 1996). To compensate for this, researchers have suggested that the timing of the first spike might convey all the information about a certain stimulus. This scheme, known as *time-to-first spike*, has been successfully applied into models for fast face recognition (Delorme and Thorpe, 2001). In a similar manner, *rank-order codes* suggest that the latency in a neuron's firing can encode all relevant information about a stimulus (Thorpe et al., 2001). This scheme uses the time that is required for a neuron to fire an action potential, by encoding a stimulus feature distributively, in the latencies of a neuronal population. Another method for translating the output of a neuron is to consider it as having on and off states during which the neuron either exhibits high firing activity or is completely shunted. This code scheme, known as the *burst code*, is a computationally convenient way for binding instantaneous features together, since it produces a succession of consecutive spikes within small periods of time.

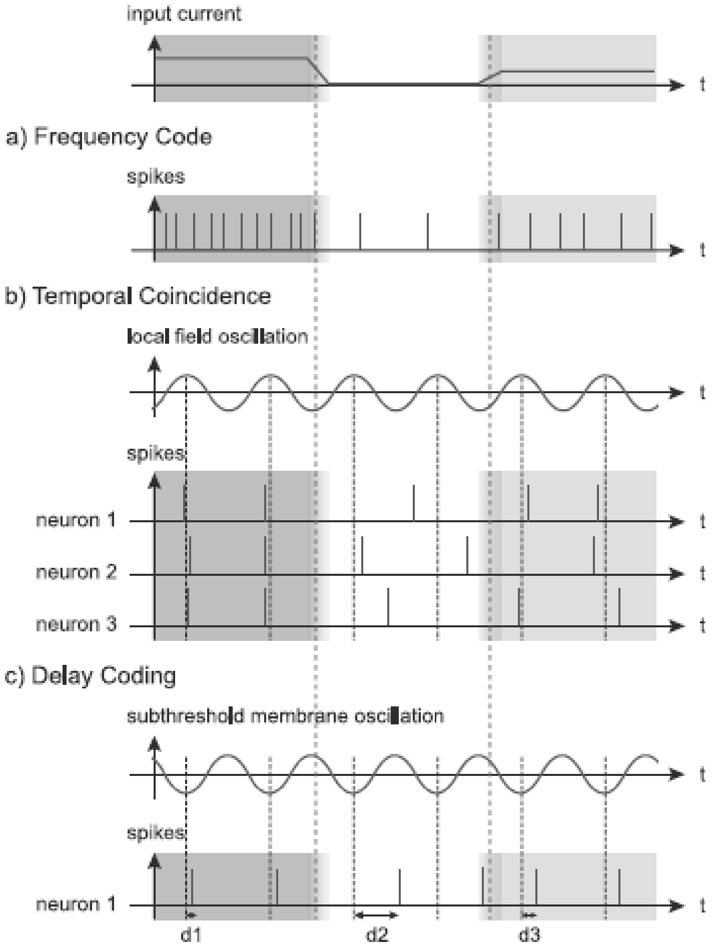


Fig. 5.2. Visual representation of 3 neural codes that can be exhibited by the spiking neuron model. Image adopted from (Florano and Mattiussi, 2008).

The above conventions consist of different interpretations of a neuron's output at the dendrite level. The choice of a neural code has a strong effect on the dynamics of a model, because it can create different forms of interactions between the neurons. In addition to the single level neural code, stimulus features can also be encoded at the population level, which is the topic discussed below.

Representations at the population level

The composition of a neural network has a strong effect on the processing capabilities of a model. By defining different neuron and synapse models, architecture and connectivity, one can build networks that perform almost any function observed in the brain. Moreover, different spatial structures can result in the network performing a completely different computation.

The nature of the neural code exhibited by a network can be described based on the distribution of the tuning curves of its neurons. Tuning curves are the most widely accepted convention for characterizing a neuron's behavior, because they describe how a cell responds to its pre-synaptic inputs. The most common type of tuning curve is the bell shaped profile, where the mean firing rate of a neuron is represented by a Gaussian function.

Depending on whether the tuning curves of its neurons overlap, a network can exhibit three different types of neural codes. On the one extreme, if a grid of neurons is characterized by tuning curves that overlap extensively, the code exhibited by the network is known as *population code* (Fig. 5.3).

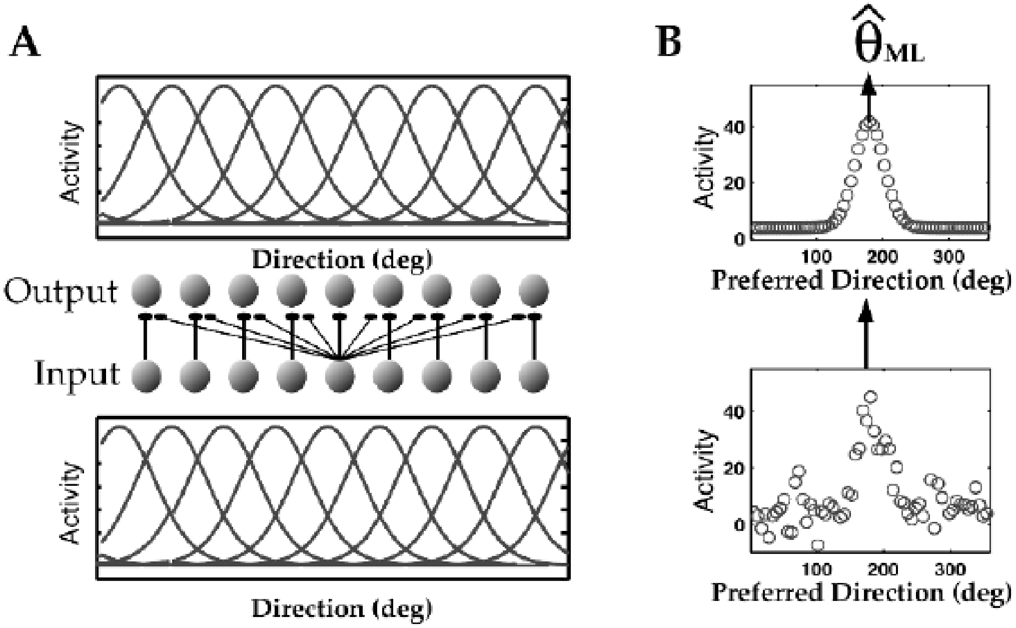


Fig. 5.3. A.The distribution of tuning curves for an ensemble of neurons that encode a population code. B. The tuning of a neuron based on a preferred, noisy, stimulus. Image adopted from (Arbib, 2003).

Population coding provides an overcomplete representation of the stimulus, using a distributed pattern of neuronal activations. Each neuron contributes, or is tuned as it is usually referred, to more than one stimulus features, while the number of neurons that is used for a representation is greater than the dimensionality of the stimulus. Population codes have been observed in many regions of the cerebral cortex, for example Georgopoulos (Georgopoulos et al., 1982) described how the properties of cell ensembles in the primary motor cortex can encode a population code vector of the monkey's hand direction and speed. This type of representation is a very good predictor of a stimulus event than the activity of any individual neuron (Schartz and Moran, 1999), however some criticize it as causing losses in the acuity of the neuronal representation (Zhang et al., 1998).

On the other end, a population of neurons can have non-overlapping receptive fields. This leads to another widely used coding scheme known as *local coding*. Local codes can divide the input space of the stimulus into segregate regions, and thus can perform better at discriminating different features. The cost for this optimized separability property is that their representational capacity is small. The number of states that can be represented by a certain population is bounded by the number of cells within the population. In the brain, local codes have been found in various regions, such as the infero-temporal cortex (Tamura and Tanaka, 2001), where researchers have identified neurons that become active only in response to specific complex shapes.

A compromising solution that takes advantage of both neural codes is *sparse coding* (Dayan and Abbott, 2001), where sensory input is encoded using a small number of active neurons. The sparseness of a population is measured by the proportion of active neurons at any given stimulus presentation (Rolls and Tovee, 1995). Usually, in a sparse code, the ratio of the number of active/inactive neurons is small. By modifying this proportion one can switch the computational capacities of a neuronal population towards optimized representation or memory. The high representational capacity of sparse codes is depicted in the fact that they have minimum entropy (Field, 1994; Barlow et al., 1989). Examples of sparseness in cortical systems can be found in the V1 cortex, which uses sparse codes in order to represent image sequences (Vinje and Gallant, 2000) or the auditory cortex of rats, where neurons produce a single spike response to a sound (DeWeese et al., 2003).

5.2.3 Models of synapses

In the current section we describe two learning methods that will enable us to develop the aforementioned neural codes in a computational agent. For this reason we focus on how two popular learning schemes, associative and reinforcement learning, can be implemented in networks of spiking neurons.

Associative learning through STDP synapses

Associating the neural representations that are encoded in different neural networks is an important function of the proposed model. Computationally, these associations are formed using the Spike-Timing Dependent Plasticity (STDP) synaptic learning rule (Song et al., 2000) which is implemented in the connections between regions SPL, IPL (IPL_{motor}, IPL_{visual} and VIP), F5 and SI. As opposed to traditional Hebbian rules, STDP ensures that: (i) non-causal relationships between neurons will not be enforced (Song et al., 2000) and (ii) correlated input activity between neurons will give rise to increased variability in the post-synaptic responses (Stevens and Zador, 1998). The above are accomplished by driving the neuron into a state where it exhibits a balanced, but irregular, firing distribution. This irregularity makes the neuron sensitive to the pre-synaptic action potentials that arrive at its membrane, even after converging to a balanced firing regime. The rule is defined by the following equation:

$$F(\Delta t) = \begin{cases} A_+ e^{\frac{\Delta t}{\tau_+}} \; if \; \Delta t < 0 \\ -A_- e^{-\frac{\Delta t}{\tau_-}} \; if \; \Delta t \geq 0 \end{cases} \quad (5.1)$$

In eq. (5.1) the timing constants τ_+, τ_- determine the time window that when Δt falls in, the synapse is either strengthened or weakened. Polarization and depression of an STDP synapse is relative to the time difference (Δt) between the firings of the pre and postsynaptic neurons. The learning rate parameters A_+, A_- determine the maximum change of synaptic modification that is allowed to occur. With the appropriate initialization of the τ_+ and τ_- timing constants, the integral of an STDP function becomes negative (Fig. 5.4), and therefore the rule presents a tendency to weaken the connection. Because of this property, non-causal coincidences (caused by pre-synaptic spiking events, occurring sporadically before and after the post-synaptic action potential) after a few cycles vanish. In contrast, pre-synaptic neurons that present a strong tendency to fire only before a post-synaptic action potential will eventually strengthen their synapses with that neuron, leading to more stable and robust associations.

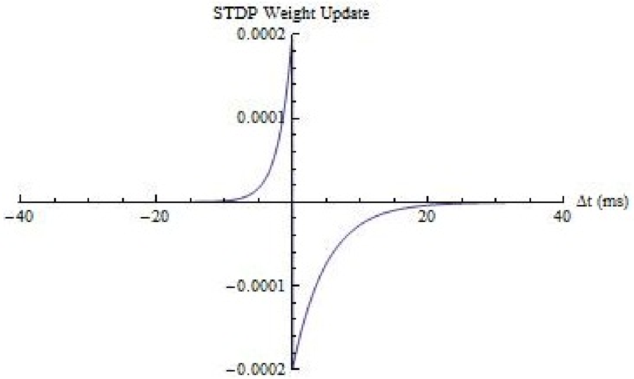


Fig. 5.4. The depression (bottom right) and polarization (top left) updates (y axis) in the weight of the STDP connections for different values of Δt (x axis). The parameters of eq. (5.1) are set to: A_+=0.0002, A_- =-0.0002, τ_+=2, τ_-=5. The choice of these parameters results in the integral of F(Δt) being negative, so that the synapse will present a tendency to weaken.

Due to these properties, STDP has been employed in a large number of computational models including ones that deal with temporal pattern recognition (Gerstner et al., 1996), coincidence detection (Gerstner et al., 1996) and directional selectivity (Mehta and Wilson, 1999). Nonetheless, STDP fails to deal with some standard problems of associative learning, such as recovering neurons that are unable to fire, due to for example small input. To confront this problem, in our simulation we have selected a relatively small value for the threshold ϑ of eq. (4.2) (see also Table 5.1), in order to ensure that on each training session a considerable number of neurons will be active. The importance of the robust associations formed due to STDP in the current model implementation becomes evident during the observation alone and observational learning phases, where the model is able to activate the same neural representations as in the observation/execution phase even though the proprioceptive input is not available. Figure 5.4 illustrates how the weight is updated relative to the time difference between a presynaptic and a postsynaptic spike, for an STDP connection with parameters: A_{-}= -0.0002, A_{+}= 0.0002 and $\tau_{+} = 2, \tau_{-} = 5$.

Associations between two regions that are connected with STDP synapses are formed by correlating neurons with similar firing frequencies. STDP accomplishes this by strengthening the synaptic connections between a presynaptic and a postsynaptic neuron if the former's firings contribute maximally to the latter's spike responses. This contribution is determined by exponentially strengthening a synapse between the two neurons, if the presynaptic neuron's firing has occurred within a small time window (determined by the τ_{-} and τ_{+} parameters) before the postsynaptic neuron's last firing. In our model STDP is used in two instances: *(i)* to form associations among the neural representations that are encoded in different neural networks and *(ii)* to promote competition between the neurons of the same networks.

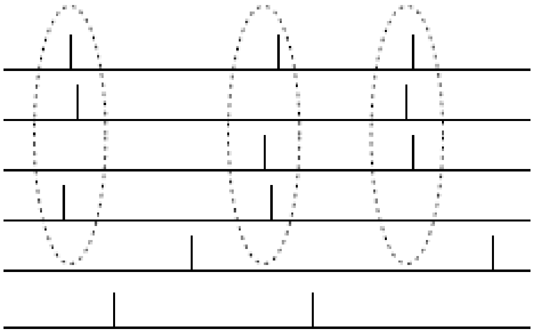


Fig. 5.5. Temporal coincidences as captures by the STDP algorithm. Circled spiking events will result in the neuron strengthening the synapse. Image adopted from (Gerstner and Kistler, 2002).

In the first case, where associations between two neural networks are formed, we use excitatory STDP synaptic connections sparsely created from the neurons of one neural network towards the neurons of another. Additional details about the parameters of these connections are given in Table 5.1.

In addition to the inter-network connectivity described above, STDP is also used to connect neurons of the same network. This second type of learning synapses, termed as *lateral-inhibitory*, is implemented as inhibitory STDP connections densely formed among the neurons of the same network (networks that employ lateral inhibitory connections are marked with a line crossing their neurons in Fig. 5.1, e.g. SPL network). The lateral-inhibitory connections ensure that the dominant firing neurons of a network will suppress the stimulation of less active cells. As a result the distributed representation encoded in each network will consist of neurons whose firing patterns concentrate more on the peaks of their tuning curves, and thus are more stable when responding to different inputs. Table 5.1 lists the neural networks that employ excitatory and lateral-inhibitory synapses, while section 5.3 discusses the type of information processing that is carried out by those networks.

Reinforcement learning

In our model new behaviours are taught through the same circuit for execution and observation, using the connections between the premotor dorsal neural networks. The synapses between those networks are adjusted in a series of observation/execution cycles (observation/execution phase), using reinforcement learning. More specifically, in this phase the agent is shown the visual depiction of an object (encoded in the V1V4$_{\text{corners}}$ and V1V4$_{\text{XYaxisRatio}}$ networks see section 5.3.1) and a certain behaviour (encoded in the STS network). This results in activating neurons in the perieto-frontal networks (through the proprioceptive association and visual object recognition pathways), which project directly to the F5$_{\text{canonical}}$ neurons. This gives rise to the initial execution of a random behaviour, which is progressively corrected in a series of observation/execution cycles using reinforcement learning based on a reward signal. The latter calculates the extent to which a behaviour generated by the motor control circuitry resembles the behaviour that was demonstrated to the agent. It is calculated on every step, for the whole presentation T$_a$=100ms of the object, using the following equation:

$$r(T_a) = \frac{\sum_{T_a}\left(M^T_{index,d} - M^T_{index,g}\right) + \sum_{T_a}\left(M^T_{thumb,d} - M^T_{thumb,g}\right) + \sum_{T_a}\left(M^T_{middle,d} - M^T_{middle,g}\right)}{3 * T_a} - 0.5 \quad (5.2)$$

where $M^T_{finger,d}$ (*finger* assumes the instances *index, middle, thumb*) is a binary reinforcement signal (0 or 1) that indicates whether the demonstrator's finger moved during the 1ms period and $M^T_{finger,g}$ a binary value that is set to 1 when the observer's corresponding finger moved during the same period and 0 otherwise. $M^T_{finger,d}$ and $M^T_{finger,g}$ are calculated every 1ms for the three fingers of our agent, and the result is summed and rescaled to the $\left[-\frac{1}{2} \cdots \frac{1}{2}\right]$ range in order to estimate the reward signal $r(t)$ used in eq. (5.3). At the end of each cycle T$_a$=100ms, this reward signal is used to compute the weight update of the reinforcement learning synapses between the neurons of the F5$_{\text{mirror}}$-F5$_{\text{canonical}}$ networks. Negative values indicate a negative reward, i.e. the generated behaviour does not resemble the demonstrated one, while positive values indicate a positive reward.

Using the reward signal from eq. (5.2), reinforcement learning is then applied in order to update the connections mentioned above. Each connection's update is derived by combining the reward signal that is generated using eq. (5.2) and an eligibility trace $z(t)$ which determines the extent of the contribution of a presynaptic neuron to the postsynaptic neuron's firing state, according to the following equation:

$$w(t) = w(t-1) + \gamma * r(t) * z(t) \quad (5.3)$$

where γ is the step size parameter, which in our model was experimentally set to the value of 0.002. Assuming that the process that generated the F5$_{\text{canonical}}$ neurons' spike trains is a point-process with probability of generating a spike $\mu(y(t), \theta)$, the eligibility trace is calculated using eq. (5.4):

$$z(t+1) = \beta * z(t) + \frac{\nabla\mu(y(t),\theta)}{\mu(y(t),\theta)} \quad (5.4)$$

In our models the bias-variance trade-off parameter β is set to 0.5. Due to the linear dynamics of the LIF model, an analytical form of the probability of generating a spike $\mu(y(t), \theta)$ can be derived from eqs. (4.1-4.5) (Baras and Meir, 2007), resulting in the eligibility trace of eq. (5.5):

$$z(t+\varDelta) = \beta * z(t) + \big(\zeta(t) - \sigma(t)\big)\frac{\lambda}{c} * \sum_{f} \exp\left(\frac{t - t_j}{\tau_m}\right) \quad (5.5)$$

where *f* denotes all presynaptic firing events (t_j) that have occurred after the last post-synaptic spike, $\zeta(t)$ is the spiking point process, λ determines the steepness of the sigmoid function and $c = \tau_m/R$ (from eq. 4.1).

5.3 Implementation of the Model pathways

In the preceding sections we have outlined the details regarding the neuron and synaptic models used throughout the networks of our model. In the current section we provide an analytic description of how all these are employed in order to construct a model of observational learning, based on the extended overlapping activations between action observation/execution. Following the design principle of pathways described in chapter 4, we start by identifying the inputs in our model, and continue to provide details about the implementation of the *(i*) object recognition, *(ii*) proprioceptive association and *(iii*) behavior learning pathways.

5.3.1 Input encoding

Our simulated agent receives three types of input: *(i)* information regarding the objects present in the scene, *(ii)* proprioceptive input that indicates the joint positions of its fingers and *(iii)* information portraying the demonstrator's finger joint positions.

For encoding the objects present in the scene, their 2D projections are acquired by means of the Webots simulator and the Matlab software is employed in order to process these images and calculate two properties which are used as identifiers of their shape: *(i)* the number of corners, and *(ii)* their XY axis ratio (For more details on how these two properties are extracted from each image refer to section 5.5.2). The two properties, when combined, are sufficient for describing the different contours of the objects (Fig. 5.10) used in our experiments. Each of these variables is encoded in a distinct neural network; the first property is encoded in the V1V4$_{corners}$ network, while the second in the V1V4$_{XYaxisRatio}$. These input networks contain tuning neurons, i.e. neurons that are tuned to respond to specific values of these two properties.

To accomplish this, each tuning neuron is pre-coded in order to acquire an average firing rate that is proportional to the difference between the network's input and its tuning value, i.e. to respond with a maximum firing rate when the input of the network is equal to its tuning value, and reduce the average firing frequency proportionally otherwise. Their membrane potential is calculated using eq. (5.6):

$$\mathrm{p(t) = p(t-1) + p_r * e^{-0.5*\frac{(a-k)^2}{\sigma^2}}} \quad (5.6)$$

At any time step t, an exponential function of the difference between the network input *k* and the neuron's tuning value *a* is added to the membrane potential *p(t)* of each neuron. The width of the tuning curve is set by the tuning sigma variable (σ). For higher values of σ, neurons respond to a larger range of inputs. Spike emission occurs when the membrane potential of the neuron exceeds the threshold value p_r. An input network encodes a certain variable using a population of tuning neurons, each adjusted to respond to a uniformly distributed range of values (using variations of the *a* and σ parameters).

This tuning neuron class is also used to construct the $\text{Sc}_{\text{proprioceptive}}$ and STS networks. These two networks encode the agent's finger joint position ($\text{Sc}_{\text{proprioceptive}}$) and the joint positions of the demonstrator (STS), respectively. Both networks comprise of three distinct sub-populations, each encoding the range of values of the joint angles for a specific finger. In the current experiments we use simulated agents with three fingers, namely index, middle and thumb. Thus each sub-population in the $\text{Sc}_{\text{proprioceptive}}$ and STS networks is responsible for encoding the joint position of its corresponding finger. The tuning values assigned to the neurons of the three sub-populations span the range of the possible joint positions for each finger in the simulator. The network input *k* for each sub-population is set on each simulation step to the position of the joints in the agent's and demonstrator's body postures, respectively.

5.3.2 Object recognition pathway

The first entry point of information in the current model is through regions $\text{V1V4}_{\text{corners}}$ and $\text{V1V4}_{\text{XYaxisRatio}}$. Those two networks are responsible for encoding the properties of the demonstrated object into population code as described in section 5.3.1. The output from those two networks is associated in region $\text{IPL}_{\text{visual}}$ which during training forms neuronal clusters in response to its inputs. The formation of clusters in $\text{IPL}_{\text{visual}}$ is accomplished by connecting neurons that are close together with excitatory links, and neurons that are distant from each other with inhibitory synapses. To determine their topological position in the network, neurons are assigned a pair of integer x,y coordinates during initialization. Neurons that have a Euclidean distance smaller than 3 units are connected with excitatory synapses, while neurons more far apart with inhibitory synapses. The weights of the excitatory synaptic links are initialized to a random value in the range of [0..0.1] while the weights of the inhibitory synapses to a random value in the range of [-0.1..0]. The learning rates (A_+, A_- parameters of eq. 5.1) of the STDP connections are set to 0.0002, following research that indicates that a small value for the learning rate of an associative connection facilitates the extraction of the first principal component of the input (Gerstner and Kistler, 2002). The time constants τ_+,τ_- of the STDP connections are set to 2 and 5, respectively, which results in a learning time window that weakens or strengthens the synapse between pre and postsynaptic neurons when their spike time difference falls into the range of [-10..0]ms and [0..20]ms, respectively. The $\text{V1V4}_{\text{corners}}$ and $\text{V1V4}_{\text{XYaxisRatio}}$ networks are densely connected to the $\text{IPL}_{\text{visual}}$

region, so that when an object is viewed by the agent more than one cluster of neurons is activated. These compete during training (through their inhibitory connections), and the dominant cluster suppresses the activation of others. To ensure that diverse objects are clustered in different topological regions in the network space of IPL_{visual}, a competition mechanism has been used in the synapses between $V1V4_{corners}$- IPL_{visual} and $V1V4_{XYAxisRatio}$- IPL_{visual}. More specifically, the weights of all synapses of a neuron in IPL_{visual} from the $V1V4_{corners}$ and $V1V4_{XYAxisRatio}$ networks are normalized in the [0..1] range. As a result, when a certain neuron in the input strengthens its connections with a specific cluster, it also suppresses the strength of the connections between that cluster and the remaining neurons in the input.

The neural code in IPL_{visual} is formed as follows. When an object is present in the scene, we extract a 2D figure from the simulator and use the Matlab software to calculate the number of corners, using the Harris and Stephens operator (Harris and Stephens, 1988) and its X/Y axis ratio (see section 5.5.2). The values extracted from the visual processing stage are input to the $V1V4_{corners}$ and $V1V4_{XYaxisRatio}$ networks, respectively. As a result, neurons in those two networks start firing with an average firing frequency that is relative to how close is their tuning value to the input. The active neurons in the two input networks subsequently activate different neurons in the IPL_{visual} network. The clusters that these neurons belong to will then compete with each other for the representation of the input network. Due to the normalization in the synapses of the IPL_{visual} neurons, the cluster that wins the competition will also suppress the strengths of the connections with the remaining input neural representations. After a small number of simulation steps, neurons in IPL_{visual} with relatively small firing frequencies will be shunted by neurons with more dominant firings and the final neural code representing the object will be formed.

5.3.3 Proprioceptive association pathway

This pathway includes regions $Sc_{proprioceptive}$, SI, SPL, $IPL_{motor,}$ VIP and STS, and is assigned two tasks: *(i)* to form the neural codes that represent the motion of the fingers of our cognitive agent (these codes are used from the motor control circuitry through the SI-MI connections and for generating a behavior through the IPL_{motor}-$F5_{mirror}$ connections) and *(ii)* to build a correspondence between the agent's and the demonstrator's actions (through the VIP-SPL circuit). An analytic description of the two functions performed by the pathway is given in the following.

The process that allows the formation of the *proprioceptive codes* of the pathway involves the $Sc_{proprioceptive}$, SI, SPL, IPL_{motor} and VIP regions. More specifically; in the developed model, $Sc_{proprioceptive}$ contains three sub-populations, each assigned to one of the index, middle and thumb fingers. The neurons in these sub-populations use the tuning neuron model described in section 5.3.1 and are assigned tuning values that span the [0..1.2] range of possible joint positions (i.e. their average firing rate is relative to the joint position of the simulated agent). $Sc_{proprioceptive}$ projects to the SI network which also contains neuron sub-populations assigned to specific fingers. Each finger neuron class of the $Sc_{proprioceptive}$ network is connected with excitatory synapses to the corresponding neuron class in the SI network. In turn each sub-population in the SI network projects to a different cluster of neurons in SPL.

Finally, SPL projects to the IPL_{motor} network which, through its connections with $F5_{mirror}$, provides information on the proprioceptive codes of the agent when generating a behavior. The fact that IPL_{motor} does not accept any connections from visual processing areas (e.g. IPL_{visual}) ensures that it will only respond to purely motor information. The neural codes formed in regions SI, SPL and IPL_{motor} become progressively sharper (i.e. less distributed and concentrating more on the peaks of their tuning curves) as they are projected from the one region to the other. In SPL and IPL_{motor}, due to the lateral inhibitory synaptic links, the neural code contains only high firing frequencies as less active neurons are shunted by more dominantly firing neurons. This helps the model to converge faster during training, as the associative processes implemented in SPL (with region VIP) and IPL_{motor} (with $F5_{mirror}$) strengthen only the synapses of dominant firing neurons. As a result, the sparse neural code formed in these regions is robust to perturbations caused by small changes in the input patterns, since any random neurons that might become active will be quickly shunted by the more dominant firing neurons. When the agent moves a finger, the value of its joint position is input to the respective sub-population of the $Sc_{proprioceptive}$ neural network. Neurons in this network with a tuning value equal or close to the current joint position will start firing with high frequency. Due to the excitatory synapses between $Sc_{proprioceptive}$ and SI, a sub-population in the SI network will also start firing when its assigned finger is active. The active SI neurons are used to recurrently provide proprioceptive information to the MI neural network for motor control (through the SI-MI connections), as well as to project to the Superior Parietal network (through the SI-SPL synapses). The term recurrently here is used to indicate the closed loop of information between the SI network and the motor control circuit (the information encoded in SI depends on the output of the motor control circuit from the previous time step, which in turn, also depends on the information from SI, due to the SI-MI connections, in order to generate the movement of the agent for the next time-step). The SPL network is sparsely (with a measure of 40%) interconnected to the IPL_{motor} network (i.e. a neuron from SPL is connected to 60% of the neurons in IPL_{motor}), which holds a distributed representation of the motion of the active fingers.

In addition to the formation of the proprioceptive codes, the proprioceptive association pathway is also responsible for the *action association* function. This is accomplished using the SI-SPL-VIP circuitry as follows: SPL, apart from SI, also accepts connections from VIP, i.e. the region encoding a distributed representation of the demonstrator's active fingers. These synapses (VIP-SPL) undergo a competition process, which aims at associating the neural representations of the agent's fingers with the neural representation formed for the demonstrator's fingers. More specifically, during training, the joint values of the demonstrator's fingers are input in the STS neural network. STS projects directly to VIP, with excitatory STDP synapses, and as a result the STS-VIP circuit encodes a distributed neural representation of the perceived motion of the fingers of the demonstrator. Therefore, the role of VIP in the current implementation is to hold a neural representation of how the model represents the actions of the demonstrator. To associate this representation with the actions of the agent, the model uses the connections between VIP and SPL (i.e. the action association is coded in the VIP-SPL synapses). By design, region SPL contains separate neuron groups that correspond to each finger in the agent's body. Each group in SPL is connected with the corresponding cluster from SI that encodes the motion of the same finger. The weights of these connections are drawn from a Gaussian distribution with mean 0.02 and sd 0.01. The connections between VIP and SPL are used in order to create an association between the local SPL code (i.e. grandmother cell representation) representing the agent's actions, and the distributed code held in VIP representing the demonstrator's actions. This association is formed due to a competition mechanism, that normalizes the weights of all synapses (in respect to the sum of their weights) leading to the same neuron in SPL from VIP in the [0..1] range. This normalization process forces the neurons in SPL to compete in order to create an association with the active VIP neurons. The neurons that win the competition (i.e. become active) in SPL, strengthen their synapses with the active neurons in VIP (due to STDP) and also weaken their connections with the inactive neurons in VIP (due to synaptic normalization). The excitatory synapses from SI to SPL will give a competitive advantage to those SPL neurons that correspond to the active fingers of the agent during the execution of a behavior. Thus after a few training cycles, and since the agent learns on every iteration to perform the demonstrated behavior better, the neurons in SPL that refer to the active fingers of the agent will become more active, and strengthen their connections with the active VIP representation. Consequently, a certain VIP representation (formed in response to the active fingers of the demonstrator) will also activate the correct combination of fingers in SPL (that refer to the corresponding fingers of the agent). This circuitry between VIP and SPL is used during observation, in order to activate the correct SPL neurons (since the agent is kept immobile and therefore there is no information from the $\text{Sc}_{\text{proprioceptive}}$ network).

5.3.4 Behavior learning pathway

The behavior learning pathway exploits information from the previous two pathways in order to observe and execute a behavior using the same networks. The entry point for the behavior learning pathway is in the premotor network which accepts connections from the following neural networks: *(i)* $\text{IPL}_{\text{motor}}$ which provides information about the motor behavior that is being executed by the agent, *(ii)* VIP network encoding the current demonstrated behavior, and *(iii)* $\text{IPL}_{\text{visual}}$ which provides information about the viewed object. The neurons in the $\text{F5}_{\text{mirror}}$ network use the same neuron model that is described in eqs. (4.1-4.4), while their synapses are updated following the STDP learning rule of eq. (5.1). The latter choice is in consistence with the associative learning hypothesis (Heyes, 2001) which suggests that mirror neurons acquire their response properties because of associative learning and the correlated experience caused by simultaneously observing and executing an action. In addition, mirror neurons have been shown to respond only to transitive actions, i.e. when an object is present in the scene and the primate executes or observes a behavior (Myowa-Yamakoshi and Matsuzawa, 1999; Umilta et al., 2001). To model this property we have used a group of LIF neurons with increased firing thresholds. Previous research has shown that using a firing threshold slightly above the mean value of the membrane potential during asynchronous input changes the computation performed by a LIF unit from linear integration of the presynaptic input, to a coincidence detector (Konig et al., 1996). Therefore, by normalizing the input current sent to the $\text{F5}_{\text{mirror}}$ neurons from the $\text{IPL}_{\text{motor}}$, $\text{IPL}_{\text{visual}}$ and VIP regions, to appropriate ranges, the neurons in the $\text{F5}_{\text{mirror}}$ network will only become active when the $\text{IPL}_{\text{visual}}$ (object present) and at least one from the $\text{IPL}_{\text{motor}}$ (executing) or VIP (observing) networks is active. This concept is depicted mathematically in the following equation:

$$I_{F5mirror,i}(t) = \frac{\sum_{AIPmotor,j} w_{i,j} S_j(t)}{4 * N_j} + \frac{\sum_{AIPvisual,k} w_{i,k} S_k(t)}{2 * N_k} + \frac{\sum_{VIP,o} w_{i,o} S_o(t)}{4 * N_o} \quad (5.7)$$

Equation (5.7) normalizes the input received from $\text{IPL}_{\text{motor}}$ and VIP networks in the $[0 \cdots 0.25]$ range and the input received from region $\text{IPL}_{\text{visual}}$ in the $[0 \cdots 0.5]$ range. In addition the firing threshold of each neuron in the $\text{F5}_{\text{mirror}}$ network is set to 0.6. As a result, the mirror neurons of our model cannot be activated from the visual representation of an object alone (since they do not respond as linear integrators), but require additionally the input from at least one of the $\text{IPL}_{\text{motor}}$ or VIP neural networks. The scaling of the input from the latter two networks to the $[0 \cdots 0.25]$ range ensures that pure input from either the $\text{IPL}_{\text{motor}}$ or VIP networks will not activate the mirror neurons. This is used in order to ensure that the behavior learning pathway will only be activated when the agent is observing or executing in the presence of an object, and thus implicitly models the mirror neurons' selectivity for transitive actions (Umilta et al., 2001). The choice of normalization constants along with the associative learning rule used to update the synapses of the $\text{F5}_{\text{mirror}}$ neurons is discussed under the spectrum of the existing theories regarding the formation of mirror neurons in section 7.2. Mirror neurons are connected to the $\text{F5}_{\text{canonical}}$ network which also receives connections from $\text{IPL}_{\text{visual}}$ region. The synaptic links between $\text{F5}_{\text{mirror}}$-$\text{F5}_{\text{canonical}}$ are updated using the reinforcement learning rule from eq. (5.3), while the synapses between $\text{F5}_{\text{canonical}}$-$\text{IPL}_{\text{visual}}$ using the STDP connections from eq. (5.1).

5.4 Motor control of the simulated agent

The $\text{Sc}_{\text{control}}$ neural network, which controls the fingers of our simulated agent, accepts neuron signals from the MI neural network which encodes in a distributed manner the input signals received from $\text{F5}_{\text{canonical}}$. The complete layout of the motor control circuitry is illustrated in Fig. 5.6. The $\text{F5}_{\text{canonical}}$ network contains three neurons, each corresponding to a specific finger in the simulated body of the agent. The $\text{F5}_{\text{canonical}}$-MI-$\text{Sc}_{\text{control}}$-SI-$\text{Sc}_{\text{proprioceptive}}$ circuitry (motor control circuitry) is evolved using genetic algorithms (the evolutionary process is described in the section 5.4.1) so that when a neuron in the $\text{F5}_{\text{canonical}}$ network is active the finger assigned to that neuron will move. In addition the Sc network is depressed during observation.

The motion of a finger in the simulated robot is preprogrammed to activate the population of neurons in the $\text{Sc}_{\text{proprioceptive}}$ network that corresponds to that finger. Neurons in those three populations of the $\text{Sc}_{\text{proprioceptive}}$ network are preprogrammed as discussed in section 5.3.3. Ten neurons are used in each population with k values spanning the [0..1.2] range, i.e. all the possible joint positions of each finger in the simulator. The $\text{Sc}_{\text{proprioceptive}}$-SI networks are connected by forming synapses between a population of neurons in the $\text{Sc}_{\text{proprioceptive}}$ network, encoding the motion of a specific finger joint, and the population of neurons in the SI network assigned to that finger joint. The SI network is then densely connected to the

MI network in order to recurrently propagate the motion information that it receives from Scproprioceptive towards MI.

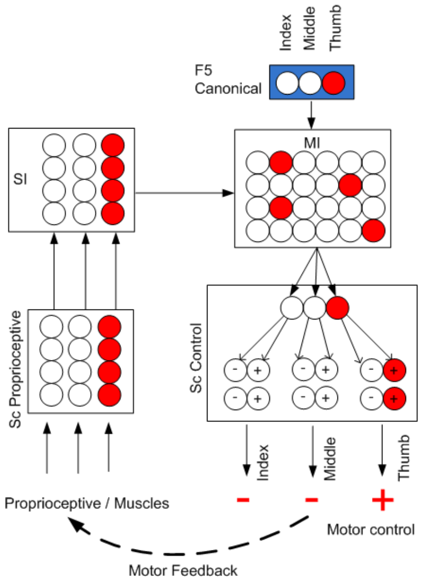


Fig. 5.6. The $F5_{canonical}$-MI-$Sc_{control}$-SI-$Sc_{proprioceptive}$ circuitry, used to control the fingers of the simulated agent. After evolution, the circuit is configured so that activations in the $F5_{canonical}$ neurons result in the respective motor commands in the $Sc_{control}$ network. Instance shown in the figure depicts the case where the thumb neuron is activated in $F5_{canonical}$ resulting in the corresponding activation in the $Sc_{control}$ network. The dashed arrow at the bottom of the figure shows how the motor commands from $Sc_{control}$ are input to the $Sc_{proprioceptive}$ network at the next step.

More details on how the signals from $Sc_{control}$ are translated to motor commands are included in section 5.4.2. The motor control circuit plans each consequent motor command using the state of the $F5_{mirror}$ neurons and the proprioceptive information from SI. When a motor command is executed from the simulator, it is subsequently encoded in the $Sc_{proprioceptive}$ network and projected to the MI network through the SI. This recurrency is important so that the agent will close its fingers up to a certain point.

5.4.1 Evolution of the motor control circuitry

Genetic algorithms are employed in order to configure the motor control circuitry. The operation of the latter is governed by a number of parameters whose operational values need to be estimated. Genetic algorithms employing simple higher order fitness functions derived from the interaction of the agent with the environment can effectively provide appropriate estimates of the sought parameters.

The training and evolutionary procedure is designed so that the motion of each finger of the agent will be controlled through the activation of a specific neuron in the $F5_{canonical}$ group of neurons. For this reason the evolutionary process exploited the parameters of the neurons in the $Sc_{control}$ and MI networks, as well as the following connections: $F5_{canonical}$-MI, MI-MI, MI-$Sc_{control}$ and SI-MI. The evaluation of each chromosome was accomplished using higher level fitness functions that rewarded in each trial the network configurations that generated a motion of a finger, if its assigned $F5_{canonical}$ neuron was active (Fig. 5.6). The chromosomes used for the evolutionary process were constructed by encoding the τ_m *, ref* and *R* parameters (from equations 4.1, 4.5) of the MI and $\text{Sc}_{\text{control}}$ neurons using an 8-bit value representation for each. In addition to that, the parameters of the connections among the $\text{F5}_{\text{canonical}}$-MI, MI-MI, MI-$\text{Sc}_{\text{control}}$ and SI-MI networks were also encoded in the chromosomes. For each of these connections we used a 1-bit value to determine whether a connection between two neurons will be formed, 1-bit for the excitation status (i.e. 1 or 0 to indicate whether the connection is excitatory or inhibitory) and an 8-bit value to determine the connection's static weight.

The encoded strings corresponding to the neuron parameters of the MI network were placed next to the encoded strings of the connections from and towards the MI network. The encoded parameters from the MI-$\text{Sc}_{\text{control}}$ connections were placed next, followed by the $\text{Sc}_{\text{control}}$ neuron parameters. The resulting chromosome string is shown in Fig. 5.7.

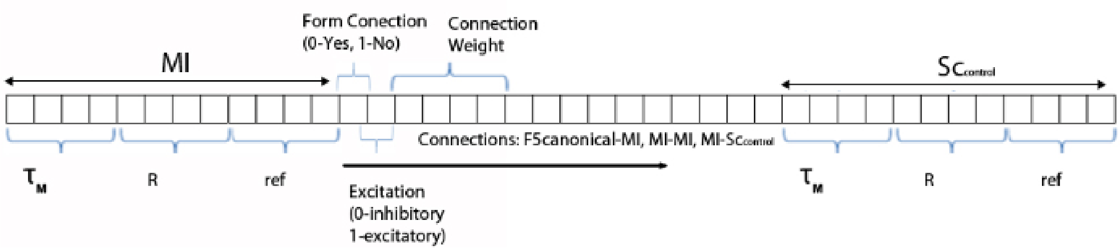


Fig. 5.7. The structure of the chromosome that encodes the neuron and connection parameters in the motor control circuit during genetic evolution.

As Fig. 5.7 shows, the chromosome string is designed to keep the encodings of a neuron's parameters close to its connections in order to minimize unwanted effects from possible separations during crossovers of the genetic evolution. Upon each evolutionary trial the neurons in the $\text{F5}_{\text{canonical}}$ network are activated sequentially for 100 steps and the motion of each finger in the body of the simulated agent is recorded. At the end of each 100 steps trial all neurons' membrane potentials are set to their resting value so that consequent 100 steps phases will be independent from each other. The circuitry $\text{F5}_{\text{canonical}}$-MI-$\text{Sc}_{\text{control}}$-SI-$\text{Sc}_{\text{proprioceptive}}$ is evaluated upon each generation, by summing the motion of the finger that is associated with the active neuron in the $\text{F5}_{\text{canonical}}$ network and subtracting the motion of the other fingers. For example during the first 100 steps of the first trial, we activate the first neuron in the $\text{F5}_{\text{canonical}}$ network and calculate the fitness of the networks according to:

$$f_{finger1} = \sum_{100} M_{finger1} - \sum_{100} M_{finger2} - \sum_{100} M_{finger3} \quad (5.8)$$

$M_{\text{finger},i}$ corresponds to the transformation of the joint of finger *i* during the 5ms period and is calculated by subtracting the last position of a joint from its current position. Each trial consists of 700 simulation steps. The neurons in the $\text{F5}_{\text{canonical}}$ network are activated separately for the first 300 steps, while during the remaining 400 steps we activate combinations of the fingers. The fitness function used in this second case is:

$$f_{finger1,2,3} = \sum_{100} \left(M_{finger1} + M_{finger2} + M_{finger3} \right) \quad (5.9)$$

The final fitness for each simulation trial of 700 steps is calculated by summing the individual fitness functions from the single and combined finger motions:

$$f_{TRIAL} = f_{finger1} + f_{finger2} + f_{finger3} + f_{finger1,2} + f_{finger1,3} + f_{finger2,3} + f_{finger1,2,3} \quad (5.10)$$

At the beginning of the evolution all connection weights are randomly initialized in the [0..1] range. The connections between the neurons of the F5$_{canonical}$-MI, MI-MI, MI-Sc$_{control}$ and SI-MI are created randomly with a probability of 50% to form a connection, and 50% for this connection to be excitatory. We used two genetic operators for evolving the chromosomes shown in Fig. 5.7, mutation and single-place roulette wheel crossover. During mutation, the algorithm randomly switches the bits of the chromosomes with a probability of 2%. During crossover, each chromosome in the genetic population is evaluated and placed in an ordered sequence according to its fitness. Chromosomes with higher fitness values have a higher probability of being chosen for crossover. When selected, a random point is set in the chromosome, which divides the string in two distinct pieces, and all bits following that point are copied from the chromosome with the higher fitness to the chromosome with the lower fitness value. This training procedure was repeated for approximately 650 generations until the activation in each F5$_{canonical}$ neuron (through the motor control circuitry) resulted in moving the finger of the agent that was assigned to that neuron. The average population fitness for the 650 generations is shown in Fig. 5.8. The configuration of the most optimal individual was decoded and used to initialize the connections and neuron parameters in the motor control circuitry.

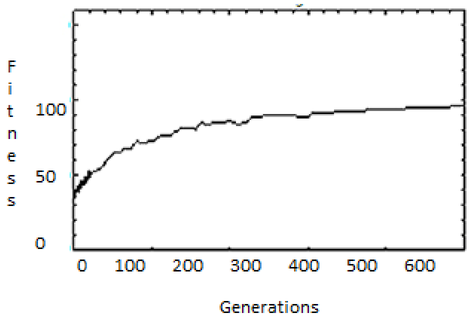


Fig. 5.8. The fitness (y axis) of the population during the initial 650 generations (x axis) of the experiment.

The accuracy of the motor control module to move the fingers of the agent to unknown positions is proportional to the distance between the designated target value, and the values used during evolution for evaluating the fitness of the algorithm. This suggests that the target positions that are used to evaluate the performance of the circuit during evolution must be carefully sampled, so that they span the target position space appropriately.

5.4.2 Control of the joints in the $Sc_{control}$ network

The joints of the simulated agent are controlled through the $Sc_{control}$ network using three populations of neurons, one assigned to each finger. Each population contains two neurons responsible for the contraction of the joint that controls the finger, and two neurons responsible for its expansion. The contribution of each neuron to the overall contraction or expansion of each joint is calculated by averaging the number of spikes it emits during a period of T=5ms according to eq. (5.11):

$$\mathrm{v} = \frac{\mathrm{n(T)}}{\mathrm{T}} \quad (5.11)$$

where $\mathrm{n(T)}$ is the number of spikes emitted by a neuron during period T. Half of the neurons in each population are responsible for the contraction of a joint and half are responsible for its expansion. The normalized sum of the average firing rates of the population controlling the expansion of the joint is subtracted from the normalized sum of the average firing rates of the population controlling its contraction. At the end of the 5ms period the result from eq. (5.12) is sent as a motor command to the simulator. $\mathrm{r_k}$ is used to determine the force that is applied to the joint of the robot.

$$\mathrm{r_k} = \frac{\sum_{\mathrm{N}} \mathrm{v_c} - \sum_{\mathrm{N}} \mathrm{v_e}}{\mathrm{N}} \quad (5.12)$$

In eq. (5.12), $\mathrm{v_c}$ represents the average firing rate for each of the N neurons controlling the opening of a joint k, while $\mathrm{v_e}$ is the average firing rate for each of the neurons controlling its closing. At the end of each 5ms period the acceleration of the joints in the simulator is set to 0 so that positive values of the $\mathrm{r_k}$ will result in the robot opening the finger, while negative values result in closing it. Figure 5.9 illustrates how the output spikes of the $Sc_{control}$ network are transformed into motor commands for the simulator.

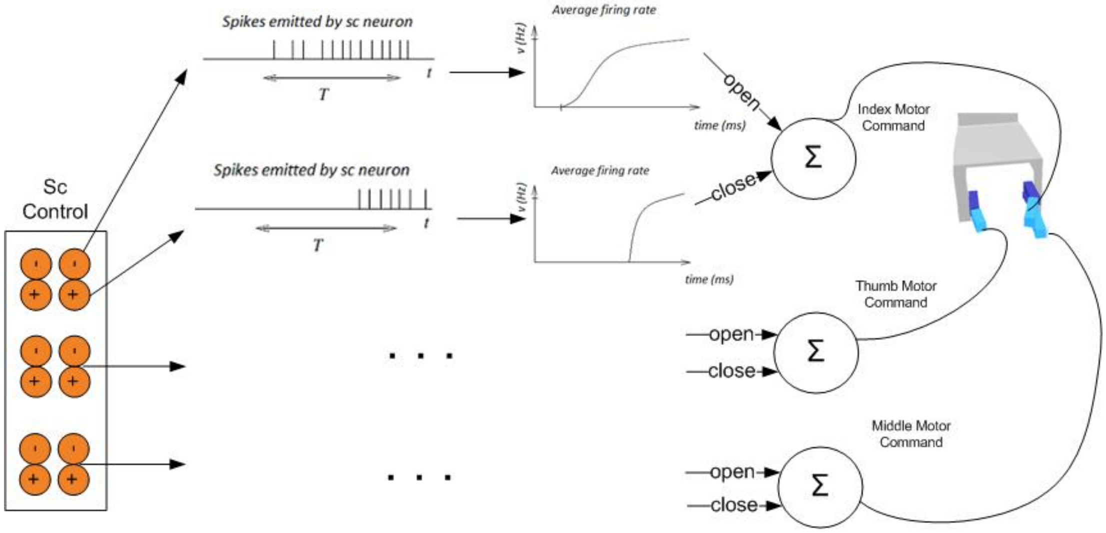


Fig 5.9. Control of the fingers in the hand component of our simulated agent through the Sc network

5.5 Model implementation details and domain applicability of the variables used in the model

In the current section we provide supplementary implementation details regarding the development of the model. These include (i) the initialization of the neurons and synapses of the neural networks and (ii) the visual processing stage of the model.Table 5.1 lists the neuron, synapse and network parameters.

Table 5.1. Simulation parameters for the neurons, connections and networks in the model

Neuron simulation parameters		**Symbol**	**Equation**	**Implementation Value**
LIF	Membrane resistance	R_m	Eq. (4.1)	10Ω
	Time constant	τ_m	Eq. (4.1)	30msec
	Reset potential	u_r	Eq. (4.3)	-60mV
	Firing threshold	ϑ	Eq. (4.2)	-10mV
	Refractory period	ref	Eq. (4.5)	1msec
Input	Tuning Value	a	Eq. (5.6)	0...1.2 (simulation units)
	Tuning sigma	σ	Eq. (5.6)	4
Connection simulation parameters		**Symbol**	**Equation**	**Implementation Value**
STDP	Learning rate	A_+, A_-	Eq. (5.1)	0.0002
	Time constant	τ_+	Eq. (5.1)	2ms
	Time constant	τ_-	Eq. (5.1)	5ms
Reinforcement	Learning rate	γ	Eq. (5.3)	0.002
	Reward value	$r(t)$	Eq. (5.3)	-0.5..0.5

Neural network simulation parameters			
	Network size	**Neuron model**	**Connection type**
V1-V4$_{\text{corners}}$	10	Input neuron	None
V1-V4$_{\text{X/Yaxis ratio}}$	10	Input neuron	None
IPL$_{\text{visual}}$	20	LIF	STDP
IPL$_{\text{motor}}$	25	LIF	STDP
SPL	15	LIF	STDP (from SI)
VIP	25	LIF	STDP
STS	15	Input neuron	None
F5$_{\text{mirror}}$	15	LIF using eq. (5.7)	STDP
F5$_{\text{canonical}}$	3	LIF	Reinforcement (from F5$_{\text{mirror}}$) STDP (from IPL$_{\text{visual}}$)
MI	15	LIF	Updated by the GA
SI	30	LIF	Updated by the GA
Sc$_{\text{control}}$	12	LIF	Updated by the GA
Sc$_{\text{proprioceptive}}$	30	Input neuron	None

5.5.1 Domain Applicability of the parameters used

The neuron parameters set in eqs. (4.1-4.5) were chosen so that neurons would respond with regular spiking events, i.e. exhibit input adaptation. The reset potential u_r was set to -60mV, while the firing threshold θ to -10mV (well above u_r) in order to allow a considerable integration of pre-synaptic action potentials between firings. The refractory period ref was set to 1ms (i.e. a low value), so that the neuron would respond to different frequencies given varying input current. Finally, membrane resistance Rm was set to 10Ω, and time constant τm to 30ms, so that the effect of the input current to the membrane potential of the neuron would be compatible with the u_r and θ values above.

Learning rate of the STDP synapses was set to 0.0002, according to research that indicates that a small value of learning rate allows an associative synapse to extract the first principal component of the input. Time constants τ_+ and τ_- were set to 2ms and 5ms respectively so that the time window of synaptic modification would consider spike time differences in the range of 0..20ms, which is in accordance to the time limits chosen for the experiment cycles (100ms). The negative time constant τ_- was set to a larger value than the positive one, so that the integral of the STDP function is negative, leading to a higher depolarization than polarization of the synapse, thus preventing non-causal relationships between neurons to be enforced. The parameters for the reinforcement learning connection were set as in (Baras and Meir, 2007). For the learning rate of the reinforcement connection a value of 0.002 was used so that changes in the model caused by the reinforcement learning connections would occur in a smaller timescale than changes due to STDP.

5.5.2 Visual processing models

In this section we outline the visual input processing models used for the objects in the scene.

Harris and Stephens operator

To obtain the corners in an image, the partial derivatives (*I_x, I_y*) of the image's intensity signal (*I*) at the x, y coordinates are calculated and then smoothed with a kernel function *F*.

$$\mathrm{T} = \begin{bmatrix} \mathrm{F}(\mathrm{I_x})^2 & \mathrm{F}(\mathrm{I_x I_y}) \\ \mathrm{F}(\mathrm{I_y I_x}) & \mathrm{F}(\mathrm{I_y})^2 \end{bmatrix} \quad (5.13)$$

In our implementation we use a rotationally symmetric 2D Gaussian kernel, with standard deviation 0.5 and kernel size 3x3, for the F function due to its smoothing properties. Corners, identified as points of interest are found if the *R* value:

$$\mathrm{R} = \det(\mathrm{T}) - \mathrm{k}[\mathrm{Tr}^2(\mathrm{T})] \quad (5.14)$$

is above a certain threshold value, which in the current experiments was set to 0.3. *k* is a tuning parameter experimentally set to 0.15.

Computation of XY axis ratio for an object

The image is converted from intensity to binary using a thresholding method (Otsu, 1979). The binary image is labeled in order to find the number of connected pixels and determine the object shape in the image (Haralick, and Shapiro, 1992). Then in the labeled version of the image the x and y axis lengths are calculated by finding the length (in pixels) between the first and the last pixel in each respective direction.

5.6 Experimental setup

To model the body posture of the agent we modified the Hoap2 robotic simulator, incorporated in the Webots package (Olivier, 2004), in order to include three fingers (thumb, middle and index) which are attached to the robot's palm. For the current experiments we have used two behaviors, each associated with the motion of one or more fingers. The first behavior entailed closing of the robot's middle and thumb fingers while the index finger remained inactive, and the second behavior involved closing the index finger while maintaining inactive the remaining ones. Since Webots controls the motion of fingers via joint angles, we have consistently employed the latter to encode finger positions, control the motion of fingers and relate robot body postures.

The experimental setup consists of two simulated robots. The first is assigned the role of the demonstrator, while the second the role of the observer. The demonstrator is preprogrammed to exhibit the behavior associated with the object present in the scene. The observer either executes this behavior in parallel with the demonstrator or just observes it without moving. The whole set of experiments is divided in three phases: *(i)* observation/execution, *(ii)* observation alone and *(iii)* observational learning. Each phase is composed of cycles, i.e. a fixed number of steps during which a certain behavior is demonstrated to the simulated agent. After the completion of each cycle the simulated robot is reset to its initial position. During one observation/execution cycle the agent is required to *(i)* observe the demonstrator performing a behavior, *(ii)* move simultaneously using the same combination of fingers, and *(iii)* touch the object. To assess whether the object is touched on each trial, the robot simulator has been embellished with binary touch sensors along its fingers. After the end of the observation/execution phase the agent is taught to execute correctly different behaviors, associate these behaviors with the object present in the scene and consequently be able to execute them whenever that object is shown, without having access to the demonstrator's motion. During the observation alone and observational learning phases, the agent is still required to observe another agent performing the same act but is not allowed to move. All cycles have a fixed length of 100ms, and are partitioned to two stages, *(i)* presentation of the object and *(ii)* grasp. During the first two phases, two different objects are used, a box, and a sphere, and each is associated with a different behavior. During the observational learning phase a third, novel, object is used, which is distinctly different from the other two objects (Fig. 5.10).

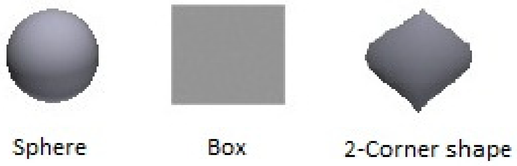


Fig. 5.10. The objects used during the observation/execution and observational learning phases. The sphere was associated with the first behavior (close middle and thumb), while the box with the second (close index finger). The third object (2-corner shape) was used during the observational learning phase.

Objects are placed in-between the agent's fingers, with all three fingers kept open at the initial step. For the sphere, the demonstrator moved the middle and thumb fingers to contact the object, while for the box the demonstrator closed its index finger (Fig. 5.11). More details about the three experiment phases are outlined below.

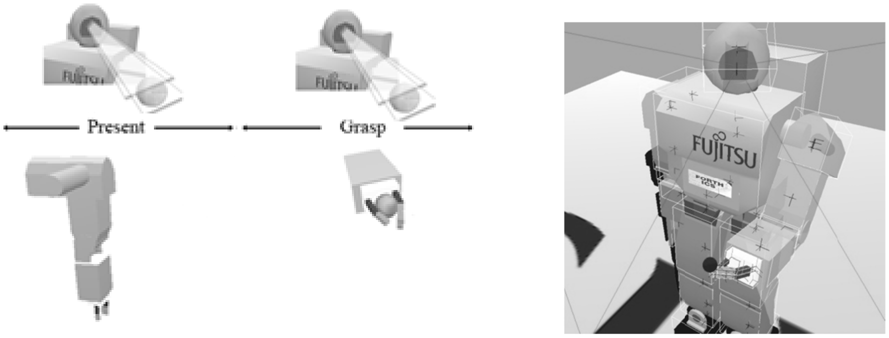


Fig. 5.11. (left) The two stages of an observation/execution phase.(right) the robot after successfully grasping an object with the appropriate combination of fingers.

5.6.1 Observation/Execution phase

During this phase the demonstrator robot exhibits two different behaviors. In the first, given a sphere object, the demonstrator closes the middle and thumb fingers. During the second, given a box object, it closes the index finger in order to touch it. Each cycle of this phase lasts for 100ms after which the simulated robot is reset to its initial position and the neurons' membrane potentials are initialized to their resting values. The membrane potentials are reset at the beginning of each cycle so that the state of the neurons will reflect the state of the robot's joints (which are also reset to a preset position at the beginning of each cycle). During the first 10ms, the observer is only shown the object in order to build the corresponding code representation in the IPL_{visual} region through the visual object recognition pathway. During the remaining 90ms, the agent is shown the behavior associated with that object, and initiates the neurons of $F5_{canonical}$ in order to execute it (Fig. 5.11). This behavior is evaluated using the reward signal of eq. (5.2), and the connections of the $F5_{canonical}$ neurons are updated through eq. (5.3) in order to correct it according to the behavior that was demonstrated to the agent. After a series of observation/execution cycles, the agent progressively rectifies the generated behavior by moving the correct combination of fingers, and thus reducing the behavior error. During training, the agent also changes the appropriate connections between VIP and SPL in order to learn the correspondence between the fingers in its own action and the action of the demonstrator.

5.6.2 Observation alone phase

During the observation alone phase the simulated agent is again shown an object and the corresponding motion of the demonstrator. However during this phase, the output of the $Sc_{control}$ neural network is not sent to the simulator and thus the neurons in the $Sc_{proprioceptive}$ network are inactive. Due to this, any top-down and bottom-up modulatory effects, caused by the $Sc_{proprioceptive}$ activation, are cancelled during observation. As a result, the activations of the model regions during observation are only attributed to the information propagated by the other two input streams (i.e. object recognition and observation of demonstrator's movement).

5.6.3 Observational learning phase

During observational learning, the simulated robotic agent is shown one of the behaviors that were taught during the observation/execution phase, and a novel object. The aim of this phase is to test whether it can learn to associate an already known behavior to a new object, without replicating it with its own simulated body, but with observation alone.

5.7 Results

In the following section we present and discuss the evaluation of the model focusing on four issues: *(i)* the learning of the two behaviors during the observation/execution phase, *(ii)* the extent to which the individual networks could be activated during the observation alone phase and whether these lower activations yielded any interesting properties regarding the learning capacities of the model, *(iii)* whether the agent could learn to associate a known behavior with a new object during the observational learning phase and whether the two behaviors taught during observation/execution are preserved after observational learning, and *(iv)* how the agent responds to unknown objects other than the ones it was trained.

5.7.1 Behavior learning

The agent learns new behaviours by observing the demonstrator. Each behaviour exhibited by the demonstrator is encoded in the STS network, while the object in the $V1V4_{corners}$ and $V1V4_{XYAxisRatio}$ networks. The input networks activate the visual object recognition, proprioceptive association and behaviour learning pathways as outlined in section 5.3. The agent executes a behaviour, and the $F5_{mirror}$-$F5_{canonical}$ connections are updated using reinforcement learning. In addition, after a few trials, a cluster starts forming in IPL_{visual} which encodes the object present on the scene and the IPL_{visual}-$F5_{canonical}$ connections strengthen. After the completion of the T_a=100ms cycle the simulator is reset to its starting position.

Reinforcement training is run for 70 cycles until the agent minimizes the error between the executed and demonstrated behaviours. The output of the training error, re-scaled to the [0..1] range, is plotted in Fig. 5.12 for the two behaviours considered during the observation/execution phase. As the figure shows, the error reaches 0 after approximately 70 cycles. At this point, the spikes on both error plots diminish, since the algorithm has converged, and the response of the $F5_{canonical}$ neurons on subsequent cycles does not require any further rectification.

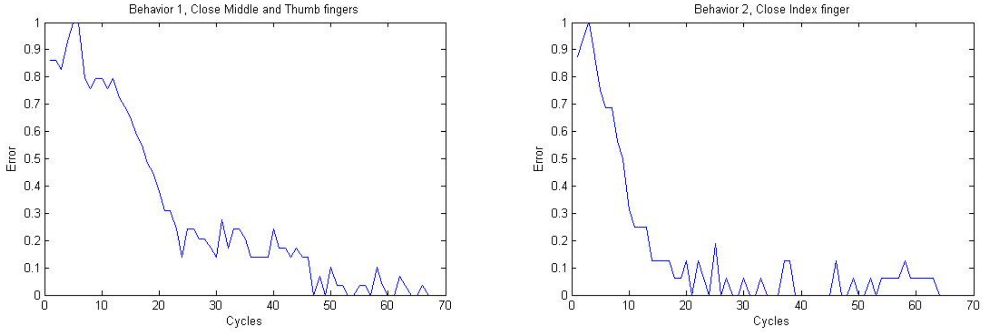


Fig. 5.12. Training error for the reinforcement learning connections during the demonstration of the first (left plot, close middle and thumb fingers) and second (right plot, close index finger) behaviour. The error signal from the three $F5_{canonical}$ neurons is summed and plotted over all trials.

After training, given a known object the agent is able to select and execute the correct behavior without any assistance from the demonstrator (i.e. the STS and VIP networks are inactive). This is accomplished as follows. When the object is shown to the agent, the cluster of neurons in IPLvisual that was formed during training in response to that object becomes active, and the F5canonical neurons that have been associated with that cluster start firing. The agent starts moving its fingers through the motor control circuit, and the proprioceptive association pathway is activated. Finally the F5mirror neurons become active (due to connections from IPLvisual and IPLmotor which are both now active) and the correct behavior unfolds using the connections from both the F5mirror and IPLvisual networks.

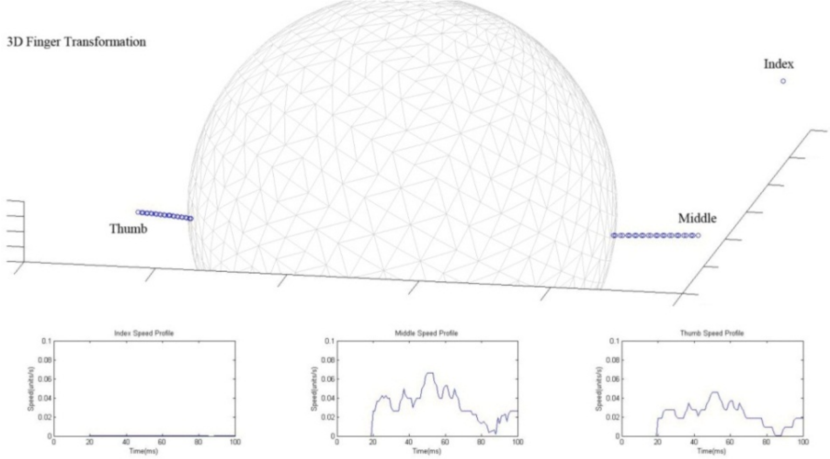


Fig. 5.13. The behaviour executed by the agent when presented with the sphere object, after the observation/execution phase. Above: A plot of the world coordinates of the three finger tips, along with a wireframe, transparent version of the object. Below: Velocity profiles for the index, middle and thumb fingers during the 100ms cycle.

In Figs 5.13 and 5.14 we show the trajectories and speed profiles of the first (Fig. 5.13) and second (Fig. 5.14) behavior, executed by the agent when the two objects that were used during training are presented.

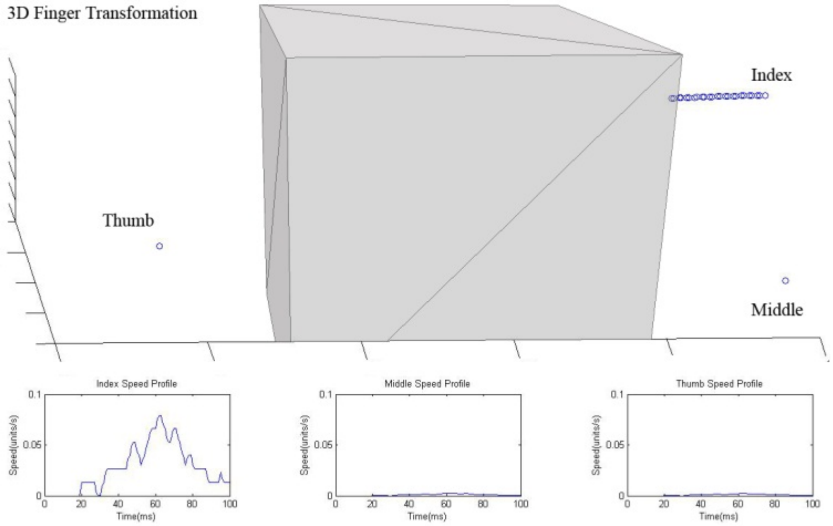


Fig. 5.14. The behaviour executed by the agent when presented with the box object, after the observation/execution phase. Above: A plot of the world coordinates of the three finger tips, along with a wireframe, transparent version of the object. Below: Velocity profiles for the index, middle and thumb fingers during the 100ms cycle.

After reaching the object, the simulated agent stops closing its fingers but still continues to exert a force for a small period. This force varies from trial to trial and depends on the speed of each finger when approaching the object.

5.7.2 Neural network activations during execution and observation cycles

As already mentioned, during the observation alone phase *(i)* the output of the $Sc_{control}$ network was not used to control the joints of the simulated robot and, *(ii)* none of the neurons in the $Sc_{proprioceptive}$ network were active. Despite this fact, the model activated regions SPL, IPL, MI and SI at a lower rate compared to the activation level during execution. This is attributed to the fact that after the successful completion of the observation/execution phase the neurons that are activated in the SPL network in response to the observation of the demonstrator’s finger motion correspond to the neurons that encode the agent’s respective finger motions. Figure 5.15 illustrates the activations during the observation/execution and observation alone phases for the four neural networks.

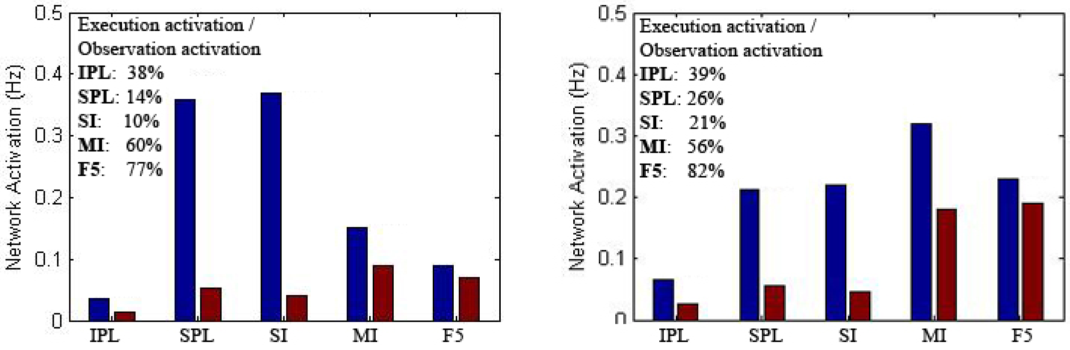


Fig. 5.15. The activations of the IPL, SPL, SI, MI and F5 networks, during the observation/execution (blue bars) and observation alone cycle (red bars). The left activation plot shows the network activations during the first behavior (close middle and thumb), while the right plot shows the network activations during the second behavior (close index). Network activations are produced by averaging all neuron spike emissions over a 100ms trial, for all neurons of a network. The legend on each plot shows the percent of activation during observation compared with the activation during execution.

Computationally we can attribute the network activations during observation alone to the active visual input of the model, originating from regions $V1V4_{corners}$, $V1V4_{XYaxisRatio}$ and STS. During observation, the agent is still shown the object and demonstrated with the associated behavior. Thus, SPL using only the synaptic links from VIP becomes active during observation, with a smaller firing rate. Consequently connections between the SPL-SI, SI-MI and SPL-IPL_{motor} networks activate the latter networks in each pair. The neurons in the subpopulations of the SI are activated at a lower rate than during execution since there is not input from the $Sc_{proprioceptive}$ network. The lower activations of the SI network result in the MI SPL and IPL_{motor} networks (which accept projections from SI) to also be activated at lower rates. Finally, since the IPL_{motor}, IPL_{visual} and VIP networks are active during both observation and execution, the $F5_{mirror}$ and $F5_{canonical}$ neurons also become active.

5.7.3 Investigation of the neuron properties during observation/execution

The results presented in the previous section indicate that the model exhibited lower regional activations during observation of a behavior as in (Raos et al., 2004). However these activations cannot be beneficial to a computational model if the corresponding neural representations that are active during observation are not consistent with the ones during execution. Moreover, to facilitate the form of observational learning we are considering, apart from activating the aforementioned networks during observation alone, it is also important to activate the appropriate neural representations that correspond to the demonstrated behavior. In the current section we look more thoroughly to the individual neuron activations in the IPL_{motor} and SPL regions of the model in order to investigate whether any informative predictions can be derived regarding the neuron activations during the observation/execution and observation alone phases.

A comparison of the active neurons during observation and execution indicates that the above mentioned networks activate the same neurons during the two phases. Consequently, the lower regional activations during the observation alone phase are caused due to some of the neurons that were active during the observation/execution phase being activated at a lower percent during observation. Figure 5.16 (plots 1, 2) illustrates the neuron activations for the SPL region, during the execution (blue bars) and observation (red bars) of the first (left plot) and second (right plot) behaviors.

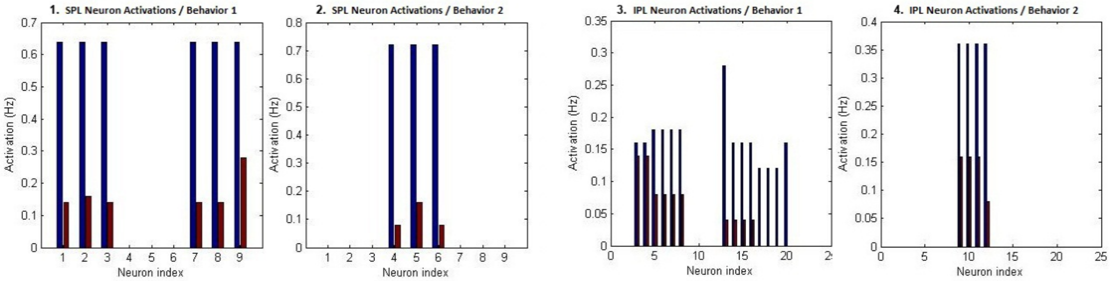


Fig. 5.16. Neuron activations for the observation/execution (blue bars) and observation alone (red bars) phases for the SPL network during the first (plot 1, close middle and thumb) and second (plot 2, close index) behavior and IPL network during the first (plot 3, close middle and thumb) and second (plot 3, close index) behavior.

As Fig. 5.16 illustrates, the SPL region activated the same pattern of neurons during execution and observation. This indicates that after training, due to the competition mechanism implemented in the connections of VIP-SPL, the neurons in SPL learn to correctly identify the code stored in VIP, and activate accordingly. This is evident from the activations shown in Fig. 5.16 (plots 1, 2), where neurons #1,2,3 correspond to the middle finger, neurons #7,8,9 correspond to the thumb finger, while neurons #4,5,6 correspond to the index finger. During execution (blue bars) and observation (red bars) of both behaviors the same combination of fingers is active. This means that the neurons in SPL that are selective to the motion of a finger of the agent, also respond (using the visual feedback from VIP) when the demonstrator is moving the respective finger. Similarly, the same neurons are also activated in IPL_{motor}. Figure 5.16 (plots 3, 4) illustrates this by plotting the activations of the IPL_{motor} region, during the execution (blue bars) and observation (red bars) of the first (left) and second (right) behaviors.

5.7.4 Observational learning

During this phase the goal is to assess the ability of the agent to associate a novel object with one of the two behaviors taught during the observation/execution phase and subsequently be able to execute this behavior whenever this object is presented. This learning process is implemented in the connections between $F5_{mirror}$-F5 $_{canonical}$ and IPL_{visual}-$F5_{canonical}$ in the behavior learning pathway. The main difference between the observational learning and the observation/execution phase is that during the former, the agent is not allowed to move.

As already mentioned, in the current phase, the agent is shown a novel object and at the same time demonstrated a known behavior, but is not allowed to move. Therefore the STS, $V1V4_{corners}$ and

$V1V4_{XYaxisRatio}$ regions are activated. Using the synaptic links between VIP and SPL, the agent knows how to activate the neurons in SPL that correspond to the active fingers of the demonstrator, and consequently all the remaining networks as outlined in section 5.3. These activations project through the IPL_{motor} network to the $F5_{mirror}$ network.

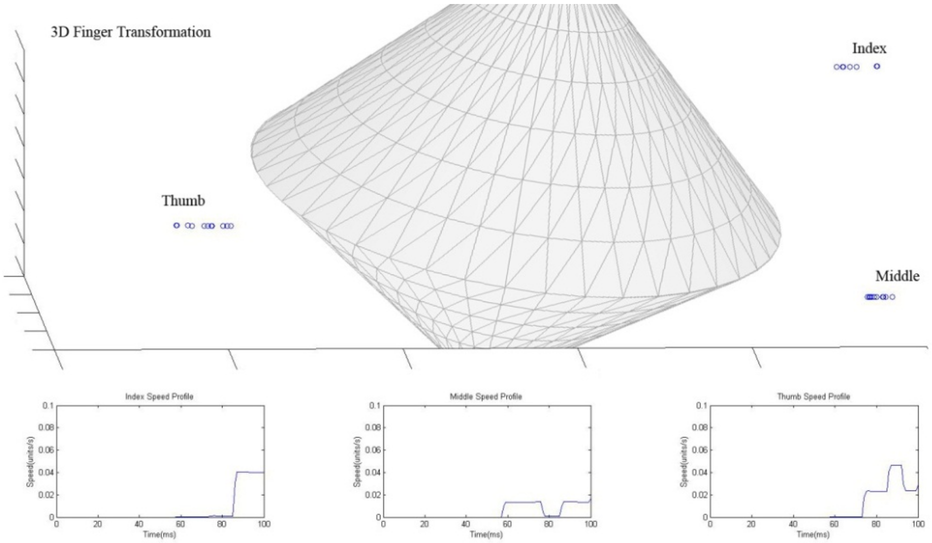


Fig. 5.17. Trajectories and speed profiles of the three fingers of the agent in response to a novel object before the observational learning phase. Above: A plot of the world coordinates of the three finger tips, along with a wireframe, transparent version of the object. Below: Velocity profiles for the index, middle and thumb fingers during the 100ms cycle.

Due to the learning during the observation/execution phase, the neurons in the $F5_{mirror}$ network that are activated are the ones that respond maximally to the behavior being demonstrated. In addition due to the new object being shown in IPL_{visual} the neuron cluster that will be activated will be associated with the active neurons in $F5_{mirror}$ and $F5_{canonical}$. After approximately 25-30 observational learning trials, the neurons with lower firing frequencies are suppressed and the only neurons that fire in the $F5_{mirror}$ network are the ones responding to the known behavior shown to the agent. These neurons build their synaptic efficacies with the neurons of IPL_{visual} that demonstrate a new object, and thus an already known behavior will be associated with a new object. Figure 5.17 illustrates the behavior that the agent executes when viewing a novel object (2-corner object of Fig. 5.10) before observational learning. As the figure shows, before observational learning the agent moves its fingers sporadically.

Subsequently we presented the 2-corner object, and programmed the demonstrator to exhibit the first behavior (close middle and thumb fingers) without allowing the agent to move. Figure 5.18 shows how the agent responds after observational learning.

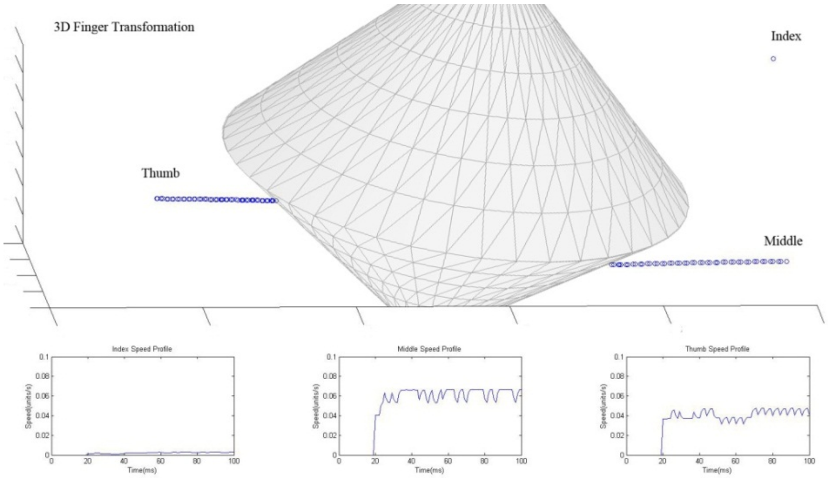


Fig. 5.18. Trajectories and speed profiles of the three fingers of the agent in response to a novel object after the observational learning phase. As the figure illustrates, the agent has learned to associate the first behaviour with a novel object only by observation. Above: A plot of the world coordinates of the three finger tips, along with a wireframe, transparent version of the object. Below: Velocity profiles for the index, middle and thumb fingers during the 100ms cycle.

As Fig. 5.18 illustrates, after the observational learning stage, when the new object is shown to the agent, the latter activates only the middle and thumb fingers. This indicates that the agent has learned to associate a previously taught behavior with a new object only by observation.

Finally, we note that after the observational learning phase, the agent is still able to execute the two behaviours learned during the observation/execution phase. This is due to the fact that in the $\text{IPL}_{\text{visual}}$ network, different objects activate different clusters of neurons. Thus when the novel object is presented during the observational learning stage, it activates a different cluster in the $\text{IPL}_{\text{visual}}$ network than the other two objects. Consequently, learning of the new behaviour during the observational learning stage will employ a different set of synapses between the IPLvisual and F5canonical networks and will not interfere with the synapses used in previous behaviours.

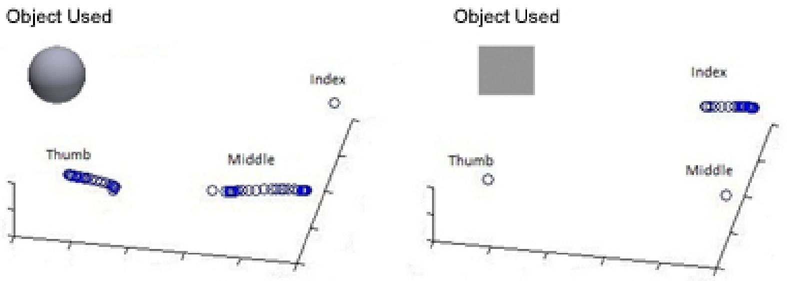


Fig. 5.19. The two behaviours executed by the agent when shown the sphere (left) and box (right) objects. As the figure shows, the agent after the observational learning phase is still able to execute the two behaviours taught during the observation/execution phase.

This property allows the agent to learn to associate new or existing behaviours to novel objects, without disturbing previous knowledge, as long as the object shown each time is considerably different to previous objects. As Fig. 5.19 illustrates, after the observational learning stage, the agent is still able to execute the *(i)* close middle and thumb behaviour (left plot) when shown the sphere object, and the *(ii)* close index finger (right plot) when shown the box object.

5.7.5 Generalization abilities of the agent

In addition to the evaluation of the agent's capacity to execute the correct behaviour, we also tested its ability to generalize to unknown objects. More specifically a series of objects were shown to the agent, some similar to the ones used during the training phase, and some different. The different objects used during this stage include *(i)* a box with 5 corners and the same X/Y axis ratio, *(ii)* a box with four corners and a 2/3 of the initial axis ratio, *(iii)* a shape with 8 corners, *(iv)* an ellipsoid with 2/3 of the initial XY axis ratio and *(v)* an ellipsoid with 3 times the initial XY axis ratio (Fig. 5.20).

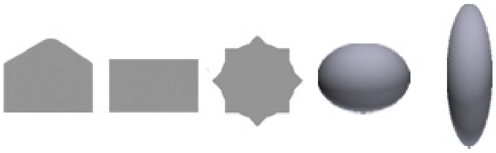


Fig. 5.20. The 5 new objects used during the testing phase.

The behaviours executed by the agent when those objects were present are shown in Table 5.2. It is evident that the agent can generalize the taught behaviors to unknown objects, i.e. is able to select the correct action even when the object that is presented is not an exact replica of the object used during training. This effect is caused due to the population coding used to encode the object properties in the input networks and the self-organization process in the $\mathrm{IPL}_{\mathrm{visual}}$ region.

One of the main properties of population coding is that neurons except from the representation that are selective to (i.e. where they exhibit a maximum firing rate activation) they also participate with lower activation to other representations. Due to the Gaussian shaped kernel (eq. 5.6) that is used for encoding the membrane potential of each cell, neurons also contribute to neighbouring values to their selective value. As a result, when for example the value of 4 is input to the $\mathrm{V1V4}_{\mathrm{corners}}$ network, to indicate an object with four corners, neurons that participate in the representations for the 5 and 3 corners also become active with lower firing frequencies. Consequently these active neurons will also strengthen their connections during training with the $\mathrm{IPL}_{\mathrm{visual}}$ neurons. This property is controlled by the σ variable of eq. (5.6), which manipulates the range of selectivity for a given neuron. If σ is excessively high, then a certain neural representation will include neurons that participate in several other representations, whereas if it is very low, the neurons active in a certain representation will not be active anywhere else. In the current simulation we have used the value of 4 for the σ parameter, which allowed us to achieve the generalization properties presented in Table 5.2.

5.7 Results

Table 5.2. The behaviours executed by the agent when different objects, than the ones trained, are presented.

Shape description	**Behaviour selected**
box with 5 corners and the same XY axis ratio	Behaviour 1
box with four corners and a 2/3 of the initial axis ratio	Behaviour 1
shape with 8 corners	No behaviour
sphere with 2/3 of the initial XY axis ratio	Behaviour 2
sphere with 3 times the initial axis ratio	No behaviour

Chapter 6

Observational learning using a phenomenological model inspired by human primates

In the current chapter we describe the development of a computational model of observational learning, loosely inspired by the neurophysiology of the human cerebral cortex, during action observation. In contrast to monkeys, humans can facilitate true imitation, i.e. the acquisition of novel motor skills. For this reason, in the current chapter, we describe an artificial agent that was developed to investigate how this ability can be embellished in a computational context, only by observation.

Due to the limited amount of data in the case of humans, the model follows a phenomenological approach, i.e. some biological constraints were loosened, in order to gain more flexibility in the behavioral design of the agent. This was accomplished only by identifying the cognitive functions that become active during observation, and suggesting a way to combine them so that the agent could exhibit observational learning skills. In this context, instead of the representation based approach that was employed in the previous chapter, we use the biologically inspired network that was described in chapter 4. In the following sections we outline the problem statement (section 6.1) and the modeling approach that is used to confront it (section 6.2). We then continue to discuss the development of the computational agent (section 6.3) and present an extensive evaluation of its ability to perform observational learning (section 6.3.3).

6.1 Problem Statement

As already discussed, the data available from human imaging experiments are much more confined compared to the corresponding ones in Macaques. This is because in humans, single cell penetrations are not permitted, and consequently the only available information pertains to higher level imaging studies that report the activation of certain regions during action observation. However, there is a close correspondence between the Macaque and human brains, and in many cases there are areas homologous to both species (Passingham, 2009). For this reason to facilitate the development of the current model, we seek inspiration from the monkey neurophysiology when more detailed data on a region are not available. Moreover, the development approach that was used in this model is different. Instead of deriving a model that is fully consistent with the cortical areas in humans, we focused on how some higher-order cognitive functions could be employed to facilitate observational learning.

In this context, and in line with the theoretical questions outlined in chapter 1 (see section 1.4), the scope of the model regards the following two issues:

1. Conclude on how the extended human motor abilities can be modeled, and how they can be employed during observation in order to derive a mental representation of the demonstrator's movement.
   In relation to theoretical question A, this issue pertains to the type of representations that will be used in order to form the mental imagined state of the agent during observation.

2. Understand how true imitation, i.e. the acquisition of novel motor skills, can be facilitated only by observation, inspired by the general functions that are performed in the regions of the human execution/observation network.
   This issue is associated with theoretical question B, and regards the learning capabilities of humans during observation, and how they can be modeled given the overlapping pathways that have been observed.

As discussed in chapter 3, humans have extended capacities for observational learning, compared to Macaques. To investigate how this can be modeled in a computational context, in the development of the agent, we also employ two additional regions the SMA and the Basal Ganglia. The functions performed in those two regions, as discussed, pertain to reward perception and higher-order motor control. Based on this intuition, in the current model we investigate how the extended overlapping pathway can facilitate observational learning of novel motor actions. In the next section we outline the modeling approach employed, which is followed by the implementation of the computational model.

6.2 Modeling Approach

To confront the two theoretical questions outlined above, we also employ the methodology of pathways. For this reason we identify the regions that become active during the observation of other human actions, and define how the activated functions can be integrated within the computational model to facilitate observational learning. In humans, one important property of their motor control system is that it develops in different stages of their life. In the first stage, humans acquire a motor control system that can:

1. Reach effortlessly towards any location.

2. Refine its target location during the execution of a movement, and follow any deviations in the trajectory.

These skills are exhibited by humans from the first stages of motor development and can be performed throughout their lives effortlessly. In addition, after having fully developed their motor control system, humans can acquire novel skills through imitation. The flexibility of their motor component suggests one important fact: they do not need to learn the underpinnings of each new behavior from scratch every time they are imitating. Instead, learning must be facilitated at a peripheral level of motor control, and employ the motor control system in order to acquire the description of new skills, based on already developed representations.

6.2.1 Highlights of the model

To facilitate the flexibility of the human neurophysiological model we must draw a clear distinction between the innate system that can perform reaching and the system that is responsible for the acquisition of novel motor skills. To accomplish this, the developed model encompasses a wider set of regions during observation, in order to facilitate learning of new motor skills without using the agent's embodiment. This constitutes a novel approach in the literature, since contemporary models of motor learning focus on imitation, i.e. the acquisition of novel skills by direct interaction.

To accomplish this, we have developed a motor control system that is structured in different hierarchical levels. On the one end, the agent is embellished with low-level motor execution modules that allow it to reach effortlessly towards any direction. At a higher-level, these modules are manipulated by a higher-order motor control component, in order to facilitate different strategies of approach towards an object. As a result of this architecture, the agent does not need to learn the underpinnings of a new motor act each time it is observing, but rather a small set of peripheral variables. This novel approach, that implements learning at the peripheral components of the motor control system, is a very important step towards robots that learn to expand their knowledge using simple variables that can be extracted only by observation.

In addition, the proposed model implements four important components of motor control, based on biologically inspired principles: *(i)* a novel primitive model that can be synthesized in order to produce complex behaviors, *(ii)* a reward assignment module, that implements the properties of the dopaminergic neurons in the Basal Ganglia, *(iii)* a higher-order motor control component that is structured on the peripheral levels of the motor system, as well as *(iv)* a state estimation module that can perform embodiment correspondence, by matching the state of the demonstrator and the observer in action space.

In the following sections we describe the implementation of the computational agent that is based on these principles. We first start by discussing the nature of the model's neural code (section 6.2.1) and continue to describe the model's pathways (section 6.3.1) as well as their implementation (section 6.3.2). The chapter is concluded by presenting an extensive evaluation of the agent's ability to learn novel motor skills only by observation (section 6.3.3).

6.2.2 The nature of the model's neural code

To complement our modeling approach, in the current implementation, we have used the Liquid State Machine that was described in chapter 4. LSMs rely on the use of biologically realistic neural networks for processing continuous input channels of information (Maass et al., 2003). Their powerful computational abilities can be attributed to the capacity of these networks to transform the temporal dynamics of an input signal into a high dimensional spatio-temporal pattern, which preserves recent and past information about the input. This information can be retrieved with a high degree of accuracy using certain types of classifiers.

The most important benefit from using Liquid State Machines is that they can preserve the input signals in the temporal domain, and thus are ideal for modeling the temporal functions associated with motor control. In addition they can capture any cognitive function, given that the properties of separation and approximation are met (Maass et al., 2003). Since the approximation property is guaranteed if one uses appropriate types of classifiers to extract information from the liquid (Legenstein and Maass., 2007), in the current model, we employ the criterion that was presented in chapter 4, in order to increase the LSM's separation. In the next section we outline the development and implementation details of the computational agent.

6.3 Computational Model of Human observational learning

Due to the fact that the body remains immobile during observation, the types of learning that can be implemented based on mental simulation have been the center of a longstanding debate in the literature. On the one end, cognitive neuroscientists (e.g. Sackett, 1934) suggest that mental practice can facilitate learning only when motor tasks include a symbolic or cognitive component, for example the association that can be formed between an object and an action. This view however is contrasted by evidence that demonstrates the improvement in physical motor performance during observation (Egstrom, 1964). On a different view, Heuer (Heuer, 1989) has postulated that mental practice has an intrinsic physical component which can be employed during observation to facilitate motor learning. Corbin (Corbin, 1967) and others (VanLehn, 1989; Sackett, 1935) have extended on this view, and suggested that mental imagery can help experienced humans to develop a mental plan of the observed movement. Learning in this case would take place in the *peripheral effects of the movement* (Corbin, 1967), by employing components that compensate for the loss of physical execution, and facilitate the acquisition of novel skills.

From the above, it is evident that in order to understand how novel skills can be acquired during motor imagery one must first identify the content of motor representations in the brain. In the current chapter we develop a computational model that can learn only by observation. For this reason, we investigate how motor control can be structured accordingly in order to enable some of its peripheral components to be optimized during observation. To accomplish this we focus on evidence that points out how the brain facilitates motor control, by decomposing the low level motor, higher-order motor control and reward components into modular systems that perform a specific function. Each modular system, which we call *computational pathway,* is identified by the regions that participate to accomplish its function and the directionality of the flow of its information. For the implementation of the cortical regions in each pathway we use Liquid State Machines.

In the rest of the chapter we describe the development of the proposed computational model of observational learning. We first derive a formal definition of observational learning in the context of computational modeling (section 6.3.1). Based on this definition, in section 6.3.2, we describe the implementation of the model, while in section 6.3.3 we show how it can facilitate learning during observation in a series of experiments that involve execution and observation of motor control tasks.

6.3.1 Definition of observational learning in the context of computational modeling

In the current section we derive the theoretical substrate for the process of observational learning in the context of computational modeling. We first focus on how the individual cognitive processes described above can be integrated together in order to support the model's function. To accomplish this we decompose the whole model into modular subsystems, which we call computational pathways (Hourdakis et al., 2011; Hourdakis and Trahanias, 2011b; Hourdakis and Trahanias, 2008), and assign specific functions to each of them. We then continue to derive a mathematical formulation of observational learning in the context of computational modeling, which is used in the subsequent section to implement the proposed model.

Pathways and modular approach to modeling

As discussed in chapter 2, to accomplish a complex task such as motor control, there are several cognitive functions that must be carried out in the cortex. To reduce the complexity of regulating all these processes, the brain makes use of modular structures (Fodor, 1982). In (Rallard and Dull, 1986) it is suggested that due to the limited number of neurons in the brain, the cerebral cortex is forced to use modular architectures. Modularity is inherent in all the stages of the cognitive processing hierarchy at both macroscopic and microscopic levels. At the lower end, cells are organized in heterogeneous circuits of increasing complexity (Hubel and Wiesel, 1962), which are subdivided in different functional areas. Each area processes a unique modality in different dimensions (Rallard and Dull, 1986) and degrees of precision (Douglas and Martin, 1998). At the macroscopic level, areas are organized into multiple parallel processing streams (a.k.a. pathways), which are responsible for carrying out a specific cognitive function (Kosslyn et al., 1990; Van Essen et al., 1992).

Modularity is also an attractive principle from a machine learning perspective. It reduces the dimensionality of a complex system into specialized building blocks, thereby confronting traditional computational problems such as cross-talk (Plaut and Hinton, 1987). In the context of motor control, the production of goal directed movements requires the coordination of interdisciplinary processes. Modularity can help towards understanding how the sensory and motor processing streams can be integrated together, by dissociating their behavioral parameters in different levels of functional processing. For this reason, to implement the task of reaching, we define segregate processing streams which we call *computational pathways* (Hourdakis et al., 2011; Hourdakis and Trahanias, 2011c; Hourdakis and Trahanias, 2008). Each pathway is assigned a distinct cognitive function and is characterized by two factors: *(i)* the regions that participate in its processing and *(ii)* the directionality of the flow of its information.

Based on the discussion in chapter 2, for the problem of observational learning of motor actions we identify six different pathways: *(i)* motor control, *(ii)* reward assignment, *(iii)* higher-order motor control, *(iv)* proprioception, *(v)* visual and *(vi)* state estimation. The functional decomposition of the model into the above modular processes will prove important in the next sub-section, where we derive a definition of observational learning in the context of computational modeling.

Mathematical derivation of observational learning

The fact that all regions that become active during execution also become active during observation suggests that when we observe, we recruit our motor component to understand an observed action. Consequently, *in terms of the motor component*, the representations evoked during mental imagery bare no difference with the ones evoked during the execution of the same action (except that in the former case, the activation of the muscles is inhibited at the lower levels of the corticospinal system). Thus to implement mental motor imagery we must first define the motor control component of the agent.

To accomplish this we adopt the definition in (Schaal et al., 2003), which suggests that when we reach towards a location, we look for a control policy π that generates the appropriate torques so that the agent moves to a desired state. This control policy is defined as:

$$v = \pi(q, t, a) \quad (6.1)$$

where v are the joint torques that must be applied to perform reaching, q is the agent's state, t stands for time and a is the parameterization of the computational model. The main difference between execution and mental imagery is that, in the latter case, the vector q is not available to the agent, since its hand is immobile. Consequently, the mental imagined state must be derived using other available sources of information. Therefore in the case of motor imagery, eq. (6.1) becomes:

$$v_o = \pi(q_o, t, a_o) \quad (6.2)$$

where q_o is the imagined state of the agent, π is as in eq. (6.1) and a_o is the parameterization of the computational model that is responsible for action observation. t denotes the time lapse of the (executed or observed) action and is not distinguished in eqs. (6.1) and (6.2) because neuroscientific evidence suggests that the time required to perform an action mentally and overtly is the same (Parsons et al., 1998). Moreover, since we operate under the neuroscientific claim that we use our motor system to simulate an observed action (Jeannerod, 1994; Roth et al., 1996), the policy π in eqs. (6.1) and (6.2) is the same. This assumption is supported by various neuroscientific evidence. Jeannerod (Jeannerod,

1988) has shown that action observation uses the same internal models that are employed during action execution. Fadiga (Fadiga et al., 1999) demonstrated that motor imagery activates the same neural pathways as in motor execution.

From the definitions of eqs. (6.1) and (6.2) we identify a clear distinction between observation and execution. In the first case the agent produces a movement and calculates its state based on the proprioception of its hand, while in the second the agent must use other sources of information to keep track of the imagined state estimate. Another dissimilarity between eqs. (6.1) and (6.2) is that the computational models a and a_o are different. However, as we discussed in chapter 3, the activations of the regions that pertain to motor control in the computational models a and a_o, overlap during observation and execution. Therefore, there is a shared sub-system in both models that is responsible for implementing the policy π (which is the same in both equations). In the following we refer to this shared sub-system as the motor control system m.

In the case of the execution computational model a, the state estimate of the agent is predicted based on the proprioceptive information of its movement, by a module p:

$$q = p(pr, m) \quad (6.3)$$

where p is the agent's internal (execution) module, pr is its proprioceptive state and m is the parameterization of its motor control system; q is the state of the agent.

During observation, the state estimate can be derived from the observation of the demonstrator's motion and the internal model of the agent's motor control system. Therefore, in the case of mental imagery of an action, the state estimate q_o is obtained by a module p_o:

$$q_o = p_o(o, m) \quad (6.4)$$

where q_o is the mentally imagined state of the agent during observation, p_o is the agent's internal (observation) model, o is the visual observation of the demonstrator's action and m is the same motor control component as in eq. (6.3). The co-occurrence of the visual observation component o and the motor control system m in eq. (6.4) constitute the basis of mental motor imagery. It states that to perform motor imagery, the computational agent must be able to integrate the information from the observation of the demonstrator with its innate motor control system. This claim is supported by neuroscientific evidence that suggests that action perception pertains to visuospatial representations, rather than purely motor ones (Chaminade et al., 2004).

The theoretical framework outlined in eqs. (6.1-6.4) describes the ground principles of our computational model. Since eqs. (6.1) and (6.2) use the same control policy π, to match the vectors v and v_o during observation and execution the agent must produce an imagined state estimate q_o that is the same as its state q would be if it was executing.

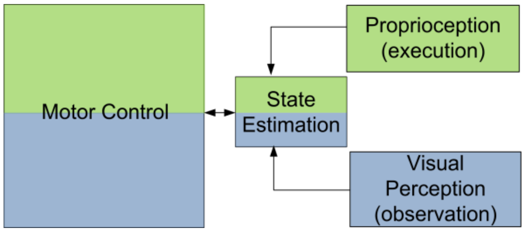


Fig. 6.1. A schematic illustration of the observation/execution system described by eqs. (6.1-6.4). The components marked in green are used during execution, while the ones marked in blue during observation. The motor control and state estimate components are used during both execution and observation.

Thus observational learning requires the implementation of an internal (observation) model p_o which will estimate the agent's imagined state q_o based on the observation stream o and the motor control module m. Equations (6.3) and (6.4) state that these estimates can be computed using the shared motor control system m of the agent. In the first case (execution, eq. 6.3), the agent must employ information from the proprioceptive pathway (pr) in order to compute its state estimate q, while in the second (observation, eq. 6.4), it must use information from the visual observation of the demonstrator (o) to compute its imagined state estimate q_0. Moreover, by examining eqs. (6.1) and (6.3) (execution), and eqs. (6.2) and (6.4) (observation) we derive another important property of the system: *To implement the policy π the motor control system must employ a state estimate, which in turn requires the motor control system in order to be computed.* This indicates a recurrent component in the circuit that must be implemented by connecting motor control with the proprioception and visual perception modules respectively.

Figure 6.1 illustrates a schematic representation of this concept, where each sub-system consists of a separate component in the computational model. The motor control and state estimation modules are shared during execution and observation, while the proprioception and visual perception modules are activated only for execution and observation respectively. In what follows we examine more closely how each individual sub-system can be implemented computationally, based on biologically inspired principles. We first examine the implementation of the motor system m, followed by the internal execution (p) and observation (p_o) models.

Reaching based on an adaptive policy

Motor control, as suggested by cognitive neuroscientists, is an integrated process that combines several different computations including higher-order motor control and monitoring of the movement. Many of our motor control skills are acquired at the early stages of infant imitation, where we learn to regulate and control our complex musculoskeletal system (Touwen, 1998). Moreover, learning of new motor skills is a developmental process that continues throughout our lives, and during which our motor system adapts and learns new control strategies.

The above suggest that learning in the human motor system is implemented at different functional levels of processing: *(i)* learn to regulate the body and reach effortlessly during the first developmental stages of an infant's life and *(ii)* adapt and learn new control strategies after the basic reaching skills have been established. This hierarchical form offers a very important benefit: one does not need to learn all the kinematic or dynamic details of a movement for each new behavior. Instead, new skills can be acquired by using the developed motor control system, and a small set of behavioral parameters that define different control strategies.

To embellish our computational agent with this flexibility in learning, its motor control system has been designed based on these principles. It consists of *(i)* an adaptive reaching component, that after an initial training phase can reach towards any given location, and *(ii)* a higher-order motor control component that implements different control strategies based on already acquired motor knowledge. In the current sub-section we outline the development of the reaching component, while in the following the development of the higher-order motor control component.

To perform reaching, the central nervous system must transform a given target location into a series of joint torques that move the end point location of the hand. In humans, neuroscientists have speculated that the target location is represented as a vector, pointing from the end position of the hand towards the object that must be reached (Gordon et al., 1994). This assumption is further supported by the evidence discussed in chapter 2 regarding the directional and force tuning of the cortical cells in the primary motor cortex. In (Todorov, 2000), Todorov has derived an almost linear local approximation between the activity of the primary motor cortex and the force activation of individual muscles. In this case, a reaching behavior is treated as a local control task where the agent produces traction forces that move its hand towards the nearest point in a given trajectory (Fig. 6.2).

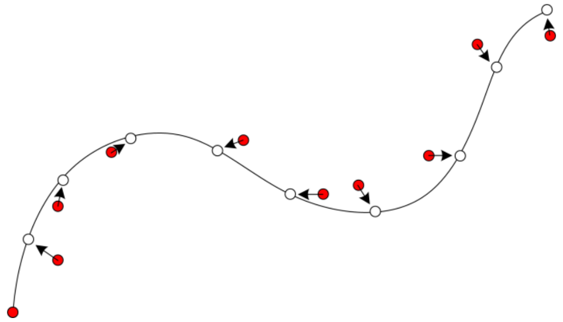


Fig. 6.2. The forces exerted by the local control policy of the motor control component and the effect they have on the movement of the plant. Red circles indicate the end position of the hand before the force is applied (the direction of the force is marked by an arrow in each position), while white circles show the target position that the hand must reach.

Computationally, such force dependent control is a difficult task due to the high dimensionality and non-linearity that is inherent in the kinematics of the multi-joint arm. Evidence from neuroscience however, suggests that at the cortical level this problem can be confronted by the use of motor primitives, i.e. low level self-organized spinal circuits that coordinate elementary motor behaviors (Thoroughman and Shadmehr, 2000). These basic motor patterns are developed during the early stages of vertebrate development where spontaneous neural activity in the spinal cord is inherent (de Vries et al., 1982).

In addition to primitives, neuroscientists have also speculated that reinforcement learning has an important role in the adaptation of this process by helping towards the formation of concrete motor patterns (Flash and Hochner, 2005). As discussed in chapter 3, the primary control center that processes reward in the brain is the Basal Ganglia, which regulates motor control through its connections with the premotor cortex. Based on this evidence, the developed computational agent is embellished with an adaptive reaching system which is implemented in the interactions of the primitive and reward assignment pathways. Section 6.3.3 presents the results of this adaptive system and demonstrates how the agent can follow any given trajectory with very good performance.

Higher-order motor control component

The second component of the motor control system pertains to a higher-order motor control. In the current paper, this is treated as an epiphenomenon of motor control and its role is to shape the trajectory when approaching the object. This is accomplished by inhibiting the forces exerted by the motor control component in a way that alters the curvature of approach towards the object (Fig. 6.3).

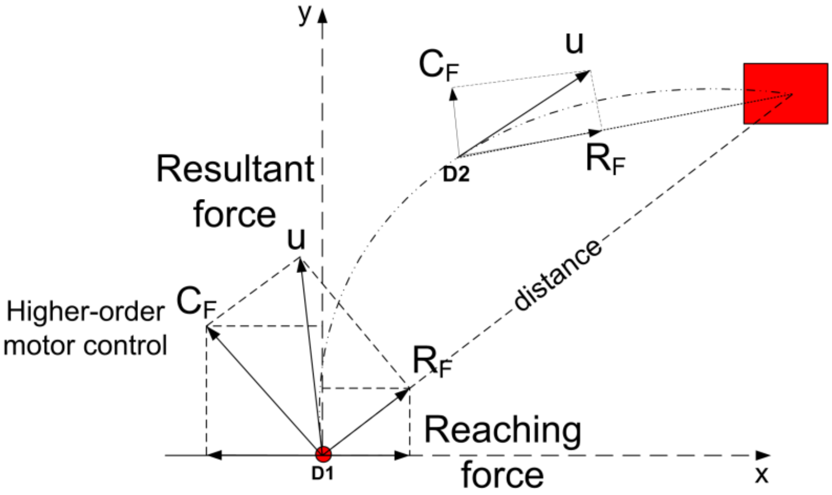


Fig. 6.3. A schematic representation of the forces that are applied to the object during reaching. The higher-order motor control component applies a force C_F, in addition to the force applied by the reaching component (R_F). The resultant force (u) changes the trajectory of the hand. In position B, the hand is closer to the object and therefore, the magnitude of the force is reduced in order to allow the reaching component to take over the motion.

The inhibition of the higher-order motor control component is realized as a force that is applied at the beginning of the movement and allows the hand to approach the target object with different trajectories (C_F force in Fig. 6.3). To ensure that the hand will reach the object in all cases, the effect of this force must converge to zero as the hand approaches the target location. This allows the reaching component to progressively take over the movement and ensure that the hand will arrive on the object. This is illustrated in Fig. 6.3, where the magnitude of the force (C_F) is reduced in position B, and therefore the reaching force (R_F) has a greater effect on the movement. In section 6.3.2 we describe the neural implementation of this component.

The fact that the higher-order motor control component depends on one single parameter, the initial force applied at the beginning of the movement, will prove important when we discuss the implementation of observational learning. The intuition is that new motor behaviors can be defined at the peripheral level of motor control, based on simple parameters that can be derived only by observation. For example in the higher-order motor control component discussed above, the initial force applied at the beginning of the trajectory could easily be inferred by the visual observation of a demonstrator without having to deal with complex aspects such as the kinematic/dynamic parameters of the observed movement. Moreover the system discussed above implements another important aspect of human motor control: the fact that humans have an innate ability to reach prior to any acquisition of novel skills or observational learning. Consequently, new skills can be taught without having to learn from scratch all the underpinnings of motor control, but a small set of simple variables instead.

Motor representations during execution and observation

During a motor control task, representations are inherent at all stages of the sensorimotor processing loop. One accepted hypothesis is that the cerebral cortex develops internal models in order to encode the extrinsic environment (Wolpert et al., 2003; Mussa-Ivaldi and Bizzi, 2000). For example, objects are encoded based on their features in neuronal populations with highly correlated properties. During perception, the role of the internal models is to remove any redundancies in these representations in order to form concrete concepts that accurately represent the input (Barlow, 1972). Cortically, the neurons that encode a certain representation are characterized by their response tuning, i.e. an increased activity in the firing rate of the cell that indicates the intensity of a certain stimulus feature (Blackemore and Cooper, 1970). Computationally this form of processing can be replicated with models of associative plasticity such as Hebbian synapses (Amari and Takeuchi, 1978) or unsupervised learning (Linsker, 1986).

In respect to observational learning, several authors have suggested that action perception is a visuo-motor event, rather than purely motor (Chaminade et al., 2004). This means that to estimate the body posture, the cerebral cortex makes use of information from proprioception and the visual input. In Macaques it was shown that there exist specific orientation cells that encode the body schema representations in an egocentric frame of reference (Perrett and Harries, 1990).

As discussed in the previous section, state estimation in our model is bi-modal, i.e. it must be dealt separately for execution and observation. However, according to eqs. (6.3) and (6.4), the state estimates that will be produced by the observation and execution models must be the same. For the first execution case, the state estimate is extracted by a feedforward neural network that is trained to calculate the end point position of the hand based on the proprioceptive input of the agent. For the observation case, the imagined state of the agent, i.e. the covert perception of the movement, is formed using also a feedforward neural network that encodes a symbolic representation of the perceived movement positions. The derivation of these components, and how these are integrated together during execution and observation, is discussed in detail in the following section, where we outline the implementation of the computational agent. Figure 6.4 shows a graphic illustration of how the system components we described in this section are mapped on the modules of Fig. 6.1.

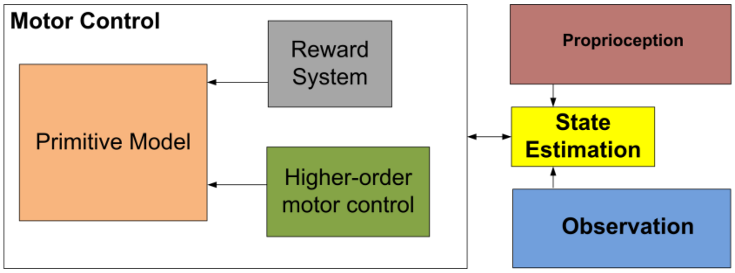


Fig. 6.4. A schematic layout of how the motor control system can interact with the proprioception and observation streams of the agent.

6.3.2 Model Implementation

Each of the separate sub-systems in Fig. 6.4 consists of a different process that is activated during motor control. To implement these processes computationally we follow biologically inspired principles. For this reason, we have carried out an analysis on how each individual cortical function is realized in the brain, and extracted the putative principles that can be used to implement it. Based on this analysis, each component in Fig. 6.4 is decomposed into several regions that contribute to its function. These are shown in Fig. 6.5, where each box corresponds to a different region in the computational agent. In this context, it is important to note that the functions identified are combined in the context of computational modeling in order to facilitate the required behavioral capacities, and do not replicate all the related cortical processes in the brain. The implementation of these regions, as well as their cortical underpinnings, are described in detail in the current section.

In Fig. 6.5 all regions are labeled based on the corresponding brain areas that perform similar functions, and are grouped according to the pathway they belong. For each of these regions (except of MI and Sc which are discussed in this section) we derive a neural implementation. For the reward assignment and higher-order motor control pathways we use Liquid State Machines (Maass et al., 2003), a recently proposed biologically inspired neural network that can process interdisciplinary functions, without requiring a circuit dependent construction. Such homogeneity is also inherent in the cortex, where stereotypical cortical microcircuits are being used for processing different functions. LSMs consist of biologically inspired neurons with locally recurrent dynamic synapses. This structure allows the neural network to preserve recent information from an input stream in the form of a spatio-temporal pattern of activation of its neurons. In the brain, similar neural activity, that acts as a short term memory storage, has been reported in all regions that we investigate (and specifically for the primary motor (Evarts and Tanji, 1976), premotor (Weinrich and Wise, 1982), somatosensory (Zhou and Fuster, 1996), supplementary motor (Tanji and Taniguchi, 1980) and visual cortex (Mikami and Kubota, 1980)). Information from the high dimensional transient states produced by an LSM can be extracted with conventional classification techniques. In (Maass et al., 2003) it was shown that LSMs can carry out any function provided that the properties of separation and approximation are fulfilled.

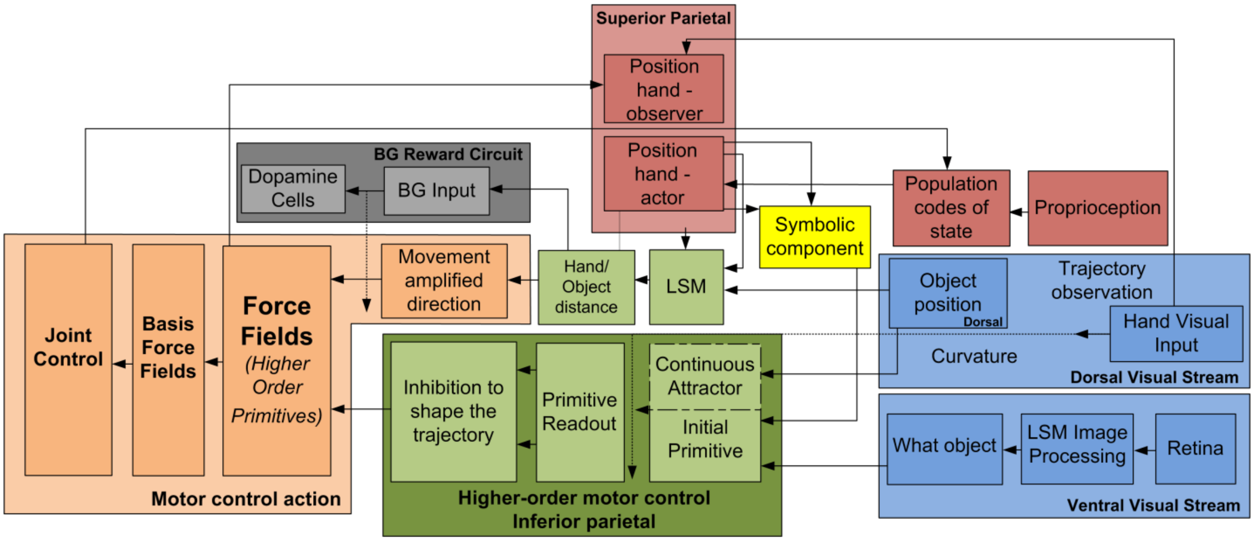


Fig. 6.5. Layout of the proposed computational model consisting of six pathways, marked in different colors: (i) visual (blue), (ii) proprioception (red), (iii) higher-order motor control (green), (iv) reward assignment (grey), (v) motor control (orange) and (vi) state estimation.

For the implementation of the visual, state estimation and forward model pathways we use feedforward neural networks and self-organizing maps which we describe later in this section. In the current section we describe the detailed derivation of each of these components. We first start by outlining the principles from control theory that will be used to produce the motion in our simulated agent, and continue to explain the implementation of each individual pathway.

Robotic Hand Control

To model the effect that the torques have on the joints of the robot we use established laws from control theory. The second order kinematics of the robot hand are modeled using the following equation:

$$D(q,\dot{q},\ddot{q}) = H(q)\ddot{q} + C(q,\dot{q})\dot{q} \quad (6.5)$$

where D is the controller that produces the torques that must be applied to the joints of the robot given its state q, and its first and second order derivatives, $\dot{q}$ and $\ddot{q}$ respectively. H is the joint-space inertia matrix and C describes the Coriolis and centripetal effects from the joint movement. Equation (6.5) can be extended with additional terms such as the viscocity of the joints or the gravity loading of the plant.

In the current implementation, we applied the model on a simulated frictionless two-link plant, and therefore, without loss of generality, we didn't include these parameters.

The aim of the computational model is to derive the appropriate local control laws that will allow the plant to reach towards any location. In practice we look for a control policy that will map the state vector of the robot to a control vector from the computational model in a way that minimizes the error of reaching. This policy is defined in eq. (6.1) for execution and eq. (6.2) for observation. The output of our model is the signal produced by the spinal cord circuit. In a biological agent the torques produced would be applied to the hand and result in movement. However since we use a simulated agent we find the second order kinematics of the hand by integrating eq. (6.5) and solving against the acceleration:

$$\ddot{\mathrm{q}} = \mathrm{H}(\mathrm{q})^{-1}\{\tau_{\mathrm{p}} - \mathrm{C}(\mathrm{q}, \dot{\mathrm{q}})\dot{\mathrm{q}}\} \quad (6.6)$$

The next configuration state of the robot is calculated using the acceleration $\ddot{\mathrm{q}}$ from the equation above, were H, C, q and $\dot{q}$ are as in eq. (6.5). The goal of the computational model is to produce the appropriate τ_p vector of joint torques that will enable the agent to perform reaching. To evaluate the proposed model we use a simulated two-link planar arm. Control is accomplished by applying torques to the elbow and shoulder joints respectively. Therefore in the presented simulations the τ_p vector is two dimensional. In the following, we first describe the computational implementation of the motor pathway, which encodes the primitive model of the agent, and subsequently, we discuss the reward assignment, state estimation and visual perception pathways.

Motor pathway

Due to the high nonlinearity and dimensionality that is inherent in controlling the arm, devising an appropriate policy for learning to reach can be quite demanding. In the current paper this policy is established upon a few higher order primitives. It turns out that, in the adopted planar arm, in order to perform any reaching behavior, only four higher order primitives are required namely up, down, left and right (Fig. 6.6). In humans such modules are formed during the first stages of the vertebrate motor development.

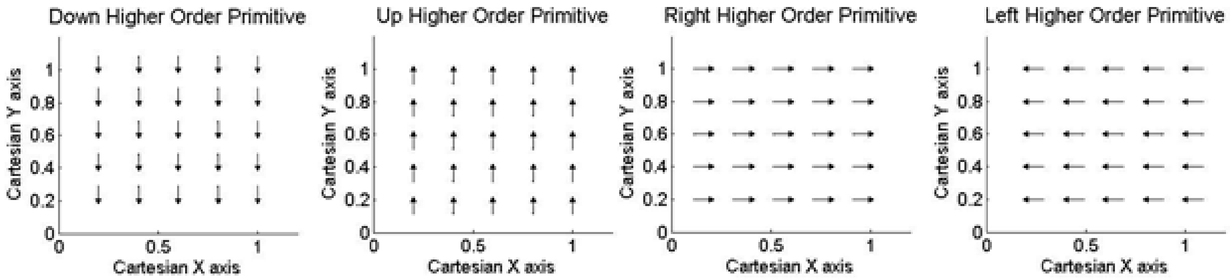


Fig. 6.6. The higher order primitive model proposed. The four plots show the force map of the primitive, i.e. the forces that are applied to the end position of the limb when the corresponding primitive is active. In the current model we use four different modules, namely up, down, left and right.

From a mathematical perspective the method of primitives, or basis functions, is an attractive way to solve the complex nonlinear dynamic equations that are required for motor control. For this reason several models have been proposed, including the VITE model that describes a way to regulate sets of agonist and antagonist muscles to move the limb to a desired state or the FLETE model that consists of a fixed parameterized system of differential equations that produce basis motor commands (see Degallier and Ijspeert, 2010 for a review). More recent studies in vertebrates suggest a force dependent encoding of motor primitives. For example experiments in paralyzed frogs revealed that limb postures are stored as convergent force fields (Bizzi et al., 1991). In (Giszter et al., 1993) the authors describe how such elementary basis fields can be used to replicate the motor control patterns of a given trajectory.

In order to make the agent generalize motor knowledge to different domains, the primitive model must be consistent with two properties: (i) superposition, i.e. the ability to combine different basis modules together and (ii) invariance, so that it can be scaled appropriately. Primitives based on force fields satisfy these properties (Gizster et al., 1993). As a result by weighting and summing the four higher order primitives shown in Fig. 6.6 we can produce any motor pattern required.

The higher order primitives are composed from a set of basis torque fields, implemented in the *Sc* module. By deriving the force fields using basis torque fields, the primitive model creates a direct mapping between the state space of the robot (i.e. joint values and torques) and the Cartesian space that the trajectory must be planned in (i.e. forces and Cartesian positions), resembling the way motions are processed by humans (Gordon et al., 1994). We first define each torque field in the workspace of the robot, and then transform it to its corresponding force field. Each torque field is described by a Gaussian multivariate potential function:

$$G(q, q_0^i) = -e^{\left(\frac{(q-q_0^i)^T K^i (q-q_0^i)}{2}\right)} \quad (6.7)$$

where q_0^i is the equilibrium configuration of each torque field, q is the robot's angle and K^i a stiffness matrix. The torque applied by the field is derived using the gradient of the potential function:

$$\tau^{\iota}(q) = \nabla G(q, q_0^i) = K^i(q - q_0^i) G(q, q_0^i) \quad (6.8)$$

Previous research has indicated that in order to achieve stability, two types of primitives must be defined: discrete and rotational (Degallier and Ijspeert, 2010). The rotational primitives are harmonic oscillators associated with a joint. The discrete ones apply a force on the hand based on a shaped valley with different equilibrium points. To ensure good convergence properties we have used 9 discrete and 9 rotational basis torque fields, spread throughout different locations of the robot's workspace (Fig. 6.7). These are generated from eq. (6.8) using different stiffness matrices. To generate the discrete torque fields (left block in Fig. 6.7) we use a semi-definite skew symmetric matrix K_{disc}, while to generate the rotational fields we use a rotation matrix, K_{rot}.

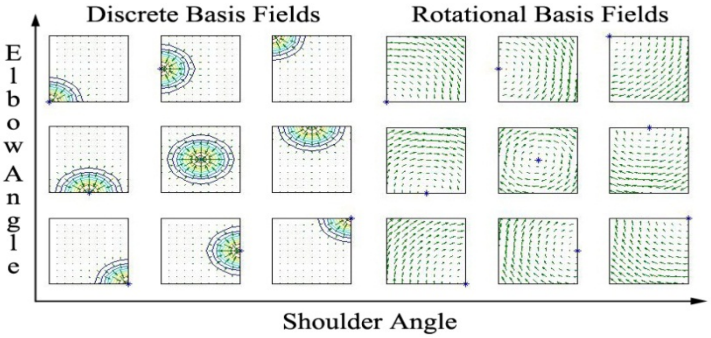


Fig. 6.7. Nine basis discrete (left block) and rotational fields (right block) scattered across the [π..π] configuration space of the robot. On each subplot the x axis represents the elbow angle of the robot while the y axis represents the shoulder angle. The two stiffness matrices used to generate the fields are $K_{disc} = \begin{bmatrix} -0.672 & 0 \\ 0 & -0.908 \end{bmatrix}$ and $K_{rot} = \begin{bmatrix} 0 & 1 \\ -1 & 0 \end{bmatrix}$.

Each plot in Fig. 6.7 shows the gradient of each torque field. The axes correspond to the $q1, q2$ joint values of the robot's hand. Since we want the model of higher order primitives to be based on the forces that act on the end point of the limb, we need to derive the appropriate torque to force transformation. To accomplish this we convert a torque field to its corresponding force field using the following equation:

$$\varphi = J^T * \tau \quad (6.9)$$

In eq. (6.9), τ is the torque produced by a torque field while φ is the corresponding force that will be acted to the end point of the plant if the torques are applied. J^T is the transpose of the robot's Jacobian. In the current implementation where the plant is located in a 2 dimensional workspace, the 6x3 Jacobian matrix can be constrained to a 2x2 matrix as:

$$J = \begin{bmatrix} -l_1 * \sin(q_1) + l_2 * \sin(q_1 + q_2) & -l_2 * \sin(q_1 + q_2) \\ l_1 * \cos(q_1) + l_2 * \cos(q_1 + q_2) & l_2 * \cos(q_1 + q_2) \end{bmatrix} \ (6.10)$$

where l_1, l_2 are the lengths of the robot's hand segments (shoulder-elbow and elbow-hand) while q_1, q_2 is the robot's elbow and shoulder angles. Each higher order force field from Fig. 6.6 is composed by summing and weighting the basis force fields from eq. (6.9). To find the weight coefficients, we form a system of N linear equations by sampling M vectors P from the robot's operational space, for all B basis force fields.

$$\begin{bmatrix} \varphi_1^1(x^1) & \cdots & \varphi_1^B(x^1) \\ \vdots & \ddots & \vdots \\ \varphi_1^1(x^M) & \cdots & \varphi_1^B(x^M) \end{bmatrix} \begin{bmatrix} a_1 \\ \vdots \\ a_M \end{bmatrix} = \begin{bmatrix} P_1^1 \\ \vdots \\ P_2^M \end{bmatrix} \quad (6.11)$$

Each higher order force field is formed by summing and scaling the basis order force fields with the weight coefficients a. The vector a is obtained from the least squares solution to the equation:

$$\Phi * \alpha = \mathrm{P} \quad (6.12)$$

In the results section we show the force fields that are produced by solving the system in eq. (6.12), as well as how the plant moves in response to a higher order force field.

Reward assignment pathway

To be able to reach adaptively, the agent must learn to manipulate its primitives using control policies that generalize across different behaviors. In the cerebral cortex one of the dominant themes used for learning is by receiving rewards from the environment. This paradigm, known as reinforcement learning in engineering, does not require an exact learning signal of the error but rather a scalar, temporally delayed, reward function (Barto, 1995). It is more consistent with the type of feedback provided to humans during learning, where exact information on the error is usually not available. An agent that learns based on reinforcement learning tries to find a policy that will maximize the probability of receiving immediate or future rewards. In the cerebral cortex, reward is processed in the dopaminergic neurons of the Basal Ganglia. One of the properties of these neurons is that they start firing when a reward is first presented to the primate, but suppress their response with repeated presentations of the same reward stimulus (Joel et al., 2002). At this convergent phase, the neurons start responding to stimuli that predicts a reinforcement, i.e. events in the near past that have occurred before the presentation of the reward.

In the early nineties, Barto (Barto, 1995) suggested an actor-critic architecture that was able to facilitate learning based on the properties of the Basal Ganglia. This architecture gave inspiration to several models that focused on replicating the properties of the dopamine neurons (see Joel et al., 2002 for a review). In the current paper we propose an implementation based on liquid state machines, and demonstrate how the interactions of this region with other neural networks of the brain can be modeled. The proposed implementation follows the actor-critic architecture and is shown in Fig. 6.8.

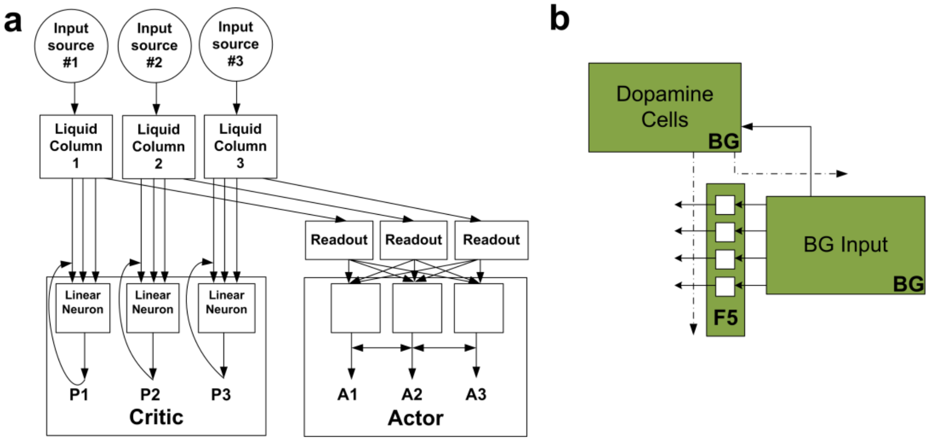


Fig. 6.8. a. The liquid state machine implementation of the actor-critic architecture. Each liquid column is implemented using a liquid state machine with feedforward delayed synapses. The critics are linear neurons, while the readouts are implemented using linear regression. b. The actor-critic architecture mapped on the model of Fig. 6.5.

The input source to the circuit consists of Poisson spike neurons that fire at an increased firing rate (above 80Hz) to indicate the presence of a certain event stimulus. Each input source projects to a different liquid column, i.e. a group of spiking neurons that is interconnected with feedforward, delayed, dynamic synapses.

The actors, i.e. the cortical region that learns based on the predicted rewards of the critics is implemented using a set of linear regression readouts that are trained to output a firing rate proportional to the sum of firing rates of each liquid column. Input from different sources is modeled as a set of rate code neurons that each projects to a separate liquid column using linear synapses with zero delay. The role of a liquid column is to transform the rate code from each input source into a spatio-temporal pattern of action potentials in the spiking neuron circuitry (Fig. 6.9).

Transmission of a spike train in each synapse is carried out with a delay of 5ms into ten neuronal layers that exist in each liquid column. This sort of connectivity facilitates the formation of a temporal representation of the input, and implicitly models the timing of the stimulus events. The occurrence of an event results in the activation of the first layer of neurons in a liquid column which is subsequently propagated towards the higher layers with a small delay. As a result, the activation of a certain neuronal layer within each liquid column represents the occurrence of a stimulus event at a particular timing interval. This temporal representation is important for the implementation of the imminence weighting scheme that is used to train the synapses of the dopaminergic Critic neurons discussed below.

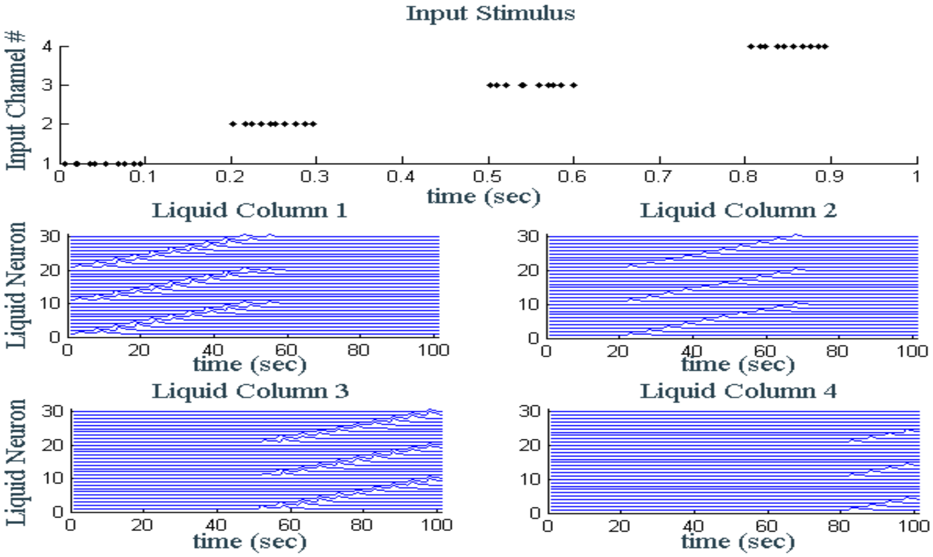


Fig. 6.9. The spatio-temporal dynamics of an event as they are transformed by a liquid column. The plot shows the temporal decay of a certain event by the liquid column's output for four stereotypical columns.

The Critic neurons $(\mathrm{P1}, \mathrm{P2}, \mathrm{P3})$ model the dopamine neurons in the Basal Ganglia. Their role is to learn to predict the reward that will be delivered to the agent in the near future. To accomplish this, the critic neurons use the temporal representation that is encoded in each liquid column and associate it with the occurrence of a reward from the environment. This is accomplished by training the synapses between the liquid columns and the critic neurons using the imminence weighting scheme (eqs. 6.13-6.15), in a way that they learn to predict the occurrence of a reward by associating events in the near past.

To implement the synapses between the liquid columns and the P, A neurons, we use the imminence weighting scheme (Barto, 1995). In this setup, the critic must learn to predict the reward of the environment using the weighted sum of past rewards:

$$\mathrm{P_t} = \mathrm{r_{t+1}} + \gamma \mathrm{r_{t+2}} + \gamma^2 \mathrm{r_{t+3}} + \cdots + \gamma^t \mathrm{r_1} \quad (6.13)$$

where the factor γ represents the weight importance of predictions in the past and $\mathrm{r_t}$ is the reward received from the environment at time t. To teach the critics to output the prediction of eq. (6.13) we update their weights using gradient learning, by incorporating the prediction from the previous step:

$$\mathrm{v_t^c} = \mathrm{v_{t-1}^c} + \mathrm{n}[\mathrm{r_t} + \gamma \mathrm{P_t} - \mathrm{P_{t-1}}]\mathrm{x_{t-1}^c} \quad (6.14)$$

where $\mathrm{v_t^c}$ is the weight of the Critic at time t, n is the learning rate and $\mathrm{x_t^c}$ is the activation of the critic at time t. The parameters γ, P and r are as in eq. (6.13). The weights of the actor are updated according to prediction signal emitted by the critic:

$$\mathrm{v_t^a} = \mathrm{v_{t-1}^a} + \mathrm{n}[\mathrm{r_t} - \mathrm{P_{t-1}}]\mathrm{x_{t-1}^a} \quad (6.15)$$

where $\mathrm{v_t^a}$ is the weight of the Actor at time t, n is the learning rate and $\mathrm{x_{t-1}^a}$ is the activation of the actor at time t-1. In the results section we demonstrate how the output of the Critic neurons approximates the response properties of the dopamine cells discussed above, as well as how the actor neurons learn to control the higher order primitive model.

The $\mathrm{A1, A2, A3}$ neurons learn based on the signal emitted by the Critic neurons. To model them in the current implementation we use a set of linear neurons. The input to each of these linear neurons consists of a set of readouts that are trained to calculate the average firing rate of each liquid column using linear regression. Each actor neuron is connected to all the readout units, with synapses that are updated using gradient descent.

The main intuition behind the proposed implementation is to transform a certain stimulus event to a temporal neural code that represents the timings of its occurrence. In section 6.3.3 we illustrate how this circuitry can replicate the properties of the dopaminergic neurons in the Basal Ganglia (discussed in chapter 2).

Visual observation pathway

The role of the visual observation pathway is to convert the iconic representation of an object into a discrete class label. This label is used by the motor control and higher-order motor control pathways in order to associate the object with specific behavioral parameters. To implement this circuitry liquid state machines were also employed. To encode the input we first sharpen each image using a Laplacian filter and consequently convolve it with 4 different Gabor filters with orientations π, $\frac{\pi}{2}$, 2π and $-\frac{\pi}{2}$ respectively (Fig. 6.10).

The four convolved images from the input are projected into four neuronal grids of 25 neurons, where each neuron corresponds to a different location in the Gabor output. Information from the liquid response is classified using a linear regression readout that is trained to output a different class label based on the object type.

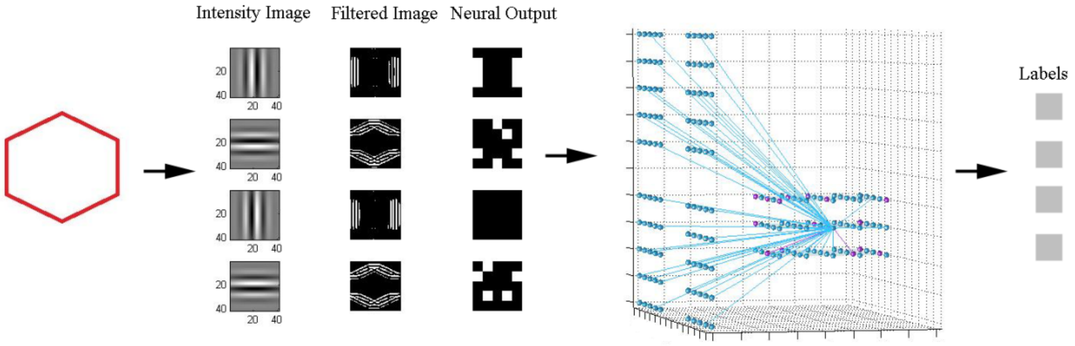


Fig. 6.10. A schematic depiction of the visual input of the agent. The iconic projection of the object is filtered, converted to neural code and input to an LSM for classification.

Forward model pathway

As discussed, one of the main transformations that take place during reaching is the cognitive implementation of a forward model (Wolpert, 1997). In the current paper, the forward model is responsible for keeping track of the execution and imagined state estimates using two different functions. The first one is implemented in the regions of the somatosensory and parietal lobe, and uses the proprioceptive input from the spinal cord in order to derive the end point position of the agent's body. During observation, the second function of the forward model is to accept information from the visual observation stream and in order to keep track of the demonstrator's end point position. In both cases the networks consist of feedforward neural nets that are trained using backpropagation.

To implement the first function we have designed the SI network to encode the proprioceptive state of the agent using population codes. This is inspired from the local receptive fields that exist in this region and the somatotopic organization of the SI (Kaas et al., 1979). Population codes assume a fixed tuning profile of the neuron, and therefore can provide a consistent representation of the encoded variable. To learn the forward transformation we train a feedforward neural network in the SPL region that learns to transform the state of the plant to a Cartesian x, y coordinate. In the results section we demonstrate how the above network is able to solve the forward kinematics problem using the agent's proprioceptive state estimates.

For the visual perception of the demonstrator's movement we have also used a feedforward neural network that inputs a noisy version (i.e. with added Gaussian white noise) of the perceived motion in an allocentric frame of reference. In this case the network is trained to transform this input into an egocentric frame of reference that represents the imagined state estimate of the observer. The noise that is used in each simulation affects the ability of the agent to correctly perceive the demonstrated behavior. In the results section we show how different noise levels can compromise the perception of the agent and, to a certain extent, its ability to learn during observation.

State estimation pathway

During observation, the role of the estimation model is to translate the demonstrator's behavior into appropriate motoric representations to use in its own system. In the literature, this problem is known as the correspondence problem between one's own and others' behaviors (Brass and Heyes, 2005). To create a mapping between the demonstrator and the imitator we use principles of self-organization, where homogenous patterns develop through competition into forming topographic maps (Udin and Fawcett, 1988). This type of neural network is ideal for developing feature encoders, because it forms clusters of neurons that respond to specific ranges of input stimuli. Furthermore, self-organization is believed to be a fundamental process in the function of perceptual systems (Rolls and Treves, 1998).

The structure of the self-organizing map (SOM) is formed, through vector quantization, during the execution phase based on the output of the forward model pathway discussed above. During its training, the network is input the end point positions that have been estimated by the forward model pathway, and translates them into discrete labels that identify different position estimates of the agent. During observation, the same network is input the computed imagined state of the visual observation model and outputs the respective labels that represent the demonstrator's movement. As discussed in the previous section, in order to implement observational learning, the outputs of the map in both cases must coincide.

For the implementation of the visual observation of the agent we have experimented with different topologies and network sizes and found that the size of the map affects the detail of the representation. Larger maps produce more detailed encodings and can represent a broader range of positions, while smaller maps discretize the input space more accurately. In the results section we demonstrate how different configurations of the SOM map affect the perception capabilities of our agent.

Reaching policy

Based on the higher order primitives and reward subsystems described above, the problem of reaching can be solved by searching for a policy that will produce the appropriate joint torques to reduce the error:

$$\mathrm{q_e} = \hat{\mathrm{q}} - \mathrm{q} \quad (6.16)$$

where $\hat{\mathrm{q}}$ is the desired state of the plant and q is its current state. In practice we do not know the exact value of this error since the agent has only information regarding the end point position of its hand and the trajectory that it must follow in Cartesian coordinates. However because our higher order primitive model is defined in Cartesian space, minimizing this error is equivalent to minimizing the distance of the plant's end point location with the nearest point in the trajectory:

$$\mathrm{d_e} = |\mathrm{l} - \mathrm{t}| \quad (6.17)$$

where l and t are the Cartesian coordinates of the hand and point in the trajectory, respectively. The transformation from eq. (6.16) to eq. (6.17) is inherently encoded in the higher order primitives discussed before. From the output of the forward model we obtain the end point Cartesian location of the hand, while from the demonstrator we obtain the point in the trajectory that must be reached. These are injected as rate codes into a liquid state machine, where a readout neuron is taught to estimate the subtraction of the two input rates using a feedforward neural network.

The policy is learned based on two elements: (i) activate the correct combination of higher order primitive force fields, and (ii) set each one's weight. Due to the binary output of the actor neurons, when a certain actor is not firing, its corresponding force field will not be activated. In contrast, when an actor is firing, its associated force field is scaled using the output of the subtraction readouts, mentioned above, and added to compose the final force.

To teach the actors the local control law, we use a square trajectory shown in Fig. 6.11, which consists of eight consecutive points $\mathrm{p}_1 .. \mathrm{p}_8$. The agent is taught the trajectory backwards, i.e. starting from the final location (p_8) in four blocks. Each block contains the whole repertoire of movements up to that point. Therefore, in the first block the actor learns to perform the left motion. Whenever it finishes a trial successfully, the actor is delivered a binary reward, and moves to the next phase which includes the movement it just learned and a new behavior.

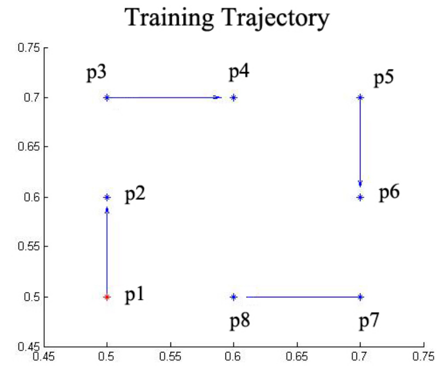


Fig. 6.11. The initial trajectory used to train the robot. It consists of 8 points that form 4 perpendicular vectors in four different directions (up, right, down, left).

Reward is delivered only when all movements in a block have been executed successfully. Therefore, the agent must learn to activate the correct force field primitives using the prediction signal from the Critic neurons in Fig. 6.8. The final torque that is applied on each joint is the linear summation of the scaled higher order primitives:

$$\tau_\mathrm{p} = \left[\mathrm{x}_{\mathrm{e},1} * (\mathrm{J}^{-1})^\mathrm{T} * \varphi_\mathrm{up}\right]_\mathrm{act1} + \left[\mathrm{x}_{\mathrm{e},2} * (\mathrm{J}^{-1})^\mathrm{T} * \varphi_\mathrm{down}\right]_\mathrm{act2} + \left[\mathrm{x}_{\mathrm{e},3} * (\mathrm{J}^{-1})^\mathrm{T} * \varphi_\mathrm{right}\right]_\mathrm{act3} + \left[\mathrm{x}_{\mathrm{e},4} * (\mathrm{J}^{-1})^\mathrm{T} * \varphi_\mathrm{left}\right]_\mathrm{act4} \quad (6.18)$$

where $x_{e,i}$ is the output from the neural network distance readout, while $[\quad]_{act}$ is an operator that includes each force field in eq. (6.18) only if the corresponding actor from the Basal Ganglia module is active. φ is obtained from eq. (6.9) for each higher order force field, and J from eq. (6.10).

Higher-order motor control pathway

Higher-order motor control is implemented at the peripheral components of the motor control system, by exerting an additional force that shapes the trajectory of the hand. As discussed in section 6.3, this force must have a large value in the beginning of the movement and reduce progressively as the hand approaches the object. When this force converges to zero, the innate reaching component of the agent takes over the motor control of the hand, and brings the end point of the plant towards the object irrespectively of its position.

To implement this concept computationally we use liquid state machines. The network consists of two liquids of 125 LIF neurons each, connected with the dynamic synapses and local connectivity (the models of neurons and dynamic synapses that were used are the same as in the reward assignment pathway). The first liquid is designed to model a dynamic continuous attractor which replicates the descending aspects of the force, while the second is used to encode the values of the additional force that will be exerted to the agent (Fig. 6.12).

To replicate the descending aspect of the force on the attractor circuit we use a liquid that inputs two sources; a continuous analog value and a discrete spike train. During training, the former, simulates the dynamics of the attractor by encoding the expected rate of descent as an analog value. The latter encodes different starting values based on the perceived starting force of the demonstrator.

The analog neuron during the tuning of the attractor is input pre-coded values that simulate the expected output, i.e. the expected rate of descent for the attractor. For the current simulations, rates of descent were sampled from three different values, 0.8, 0.1 and 0.9 respectively. During training of the circuit, the values expected by the attractor are calculated and input to the analog input neuron, which is connected to all neurons of the first liquid of the circuit. In this phase, the first liquid is simulated for 1000ms duration and its liquid states are recorded. One linear readout neuron, whose connections are setup through linear regression, is used to extract the desired values for the attractor. This is accomplished by using the input to the analog neuron as training values and finding the correct weighting coefficients to reproduce them (so that the input analog neuron will be simulated internally by the readout neuron).

The burst code neuron is programmed to fire a burst at the beginning of the simulation with duration equal to the rate of descent. For example if the rate of descent is 0.05, meaning that on each simulation step the attractor output value is expected to be reduced by 0.05, the burst code neuron will fire a burst of spikes (i.e. a Poisson spike train with average rate over 80Hz) for 5ms. To complete the training phase we gather 500 training samples of different rates of descent and 50 testing samples that are used to evaluate the attractor's performance. After training the output of the readout neuron is connected to the liquid of the attractor and the artificial analog neuron used during training is disconnected.

The second liquid in the circuit consists of 125 LIF neurons interconnected with local connectivity. The input to the network consists of three sources. The first is the readout trained by the attractor circuit that outputs the rate of descent of the movement. The second is the neuron that inputs the start force of the movement. The third source is a population code that inputs the symbolic labels from the map of the state estimation pathway.

The second neuron encodes the start force observed as a burst code. For example if the observed force is 0.1Nt then this input neuron will output a burst code (a Poisson spike train with average rate over 80Hz) for the first 100ms and maintain for the rest simulation a low activity (Poisson spike train with average rate below 10Hz). The third neuron in the circuit encodes the symbolic labels output from the state estimation pathway. The input to the circuit is the label output by the SOM module, which is encoded as a population code vector and input to the LSM. Following the 100ms temporal resolution used for the simulations during observation learning, the liquid used to train the higher-order motor control component is also simulated for 100ms, i.e. on every time-step of the simulation.

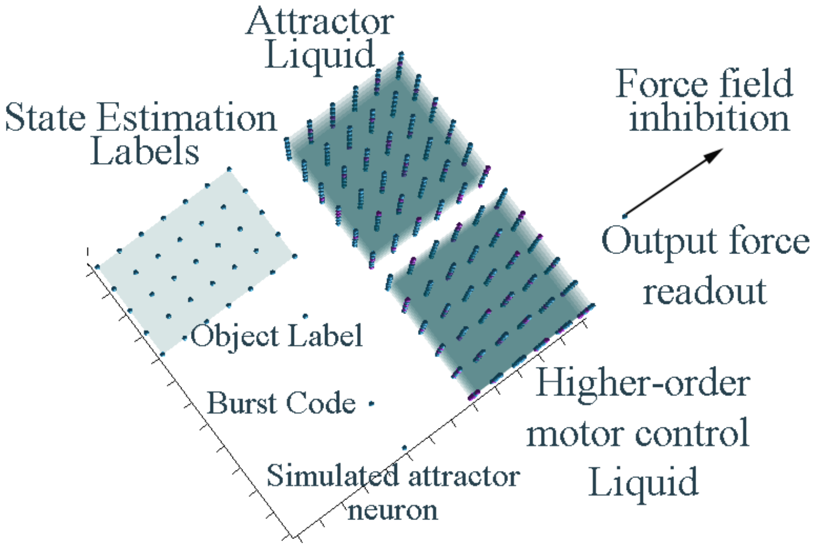


Fig. 6.12. The circuitry of the higher-order motor control pathway showing all the individual components and attractor and higher-order motor control liquids.

After collecting the states for the observer every 100ms, the liquid states are trained for every label presented to the circuit up to that moment. For example if the simulation is on the 5th step, then the input to the network is a five dimensional vector that contains the symbolic labels up to the first 500ms of the simulation. The output of these liquid states are trained using a feedforward neural network readout that must learn to approximate the start force of the demonstrator and rate of descent of its movement, based on the simulated liquid states.

6.3.3 Results

In the current section we present the results of the proposed model. We first illustrate the training results for each individual pathway of the model. We then continue to show the ability of the motor control component to perform online reaching, i.e. reach towards any location with very little training. Finally, we present the results of the observational learning process, i.e. the acquisition of new behaviors only by observation.

Motor control component training

The first result we consider is the convergence of the least squares solution for the system of linear equations in eq. (6.11). Figure 6.13 presents the solution for the "up" higher order primitive, where it is evident that the least squares algorithm has converged to a good result. The three subplots at the bottom show three snapshots of the hand while moving towards the "up" direction when this force field is active. Similar solutions were obtained for the other three primitives, where the least squares solution converged to 7 (left), 2 (right) and 5 (down) errors (the error represents the extent to which the directions of the forces in a field deviate from the direction that is defined by the primitive).

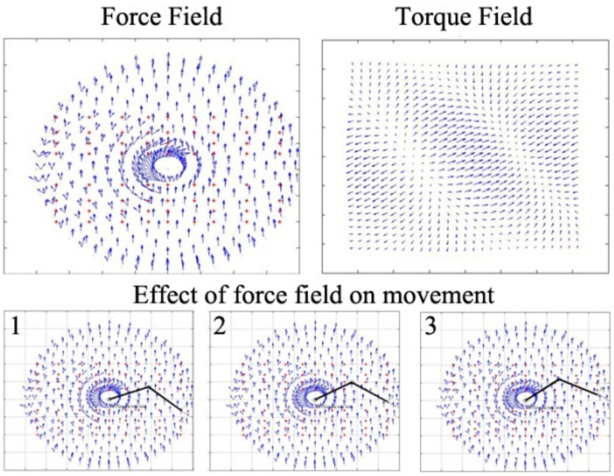


Fig. 6.13. The force field (upper left subplot) and torque field (upper right subplot) as converged by the least squares solution for the "up" primitive. The three subplots at the bottom show the snapshots of the hand while moving when the primitive is active.

Reward assignment pathway

The policy for reaching was learned during an initial imitation phase where the agent performed the training trajectory, and was delivered a binary reinforcement signal upon successful completion of a whole trial. Since the reward signal was only delivered at the end of the trial, the agent relied on the prediction of the reward signal elicited by the critic neurons. In the following we look more thoroughly in the response properties of the simulated dopaminergic critic neurons and how the actors learned to activate each force field accordingly based on this signal.

Figure 6.14 illustrates how the critic neurons of the model learned to predict the forthcoming of a reward during training. In the first subplot (first successful trial) when reward is delivered at t=4, the prediction of the 1st critic is high, to indicate the presence of the reward at that time step. After the first 10 successful trials (Fig. 6.14, subplot 2), events that precede the presentation of the reward (t=3) start eliciting some small prediction signal. This effect is more evident in the third and fourth subplots where the prediction signal is even higher at t=3 and starts responding at t=2 as well.

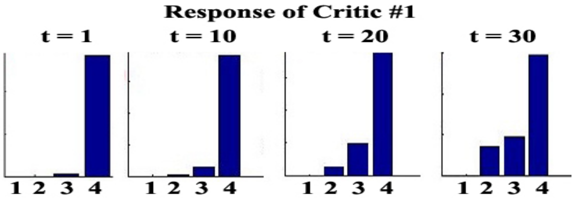


Fig. 6.14. The prediction signal emitted by the critic component of the model during the initial stages of the training (subplot 1), after 10 trials (subplot 2), after 20 trials (subplot 3) and after 30 trials (subplot 4).

The effects of this association are more evident in Fig. 6.15, where it is shown how after training, even though rewards are not available in the environment, the neurons start firing because they predict the presence of a reward in the subsequent steps. Using the output of this prediction signal, the actor, i.e. in the case of the model the F5 premotor neurons that activate the force fields in the MI, forms its weights in order to perform the required reaching actions.

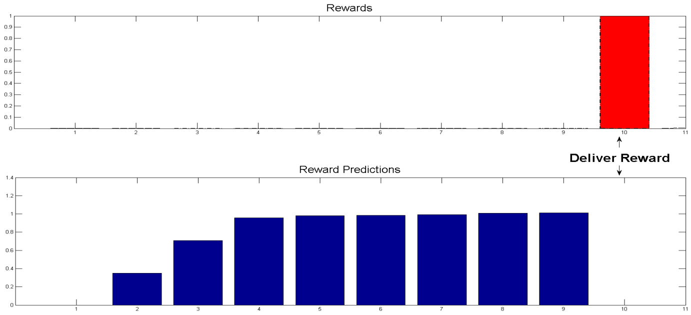


Fig. 6.15. The actual reward signal given to the robot at the end of a successful trial (upper subplot), and the reward predicted by the critic component after training (bottom subplot). The x-axis represents the 100ms time blocks of the simulation while the y-axis the values of the reward and prediction signals respectively.

Forward model

In the current section we present the results from the training of the two feedforward neural networks that were used in order to implement the forward models of the agent. In the first case, the network was trained in order to perform the forward transformation from the proprioceptive state of the agent to the end point position of its hand. For this reason the joint positions of the simulated agent were extracted in every step of the simulation and encoded as population codes in the Sc module. The feedforward neural network consisted of two layers of sigmoidal activation neurons, and was trained for 100 iterations, to output the end point position of the hand (Fig. 6.16).

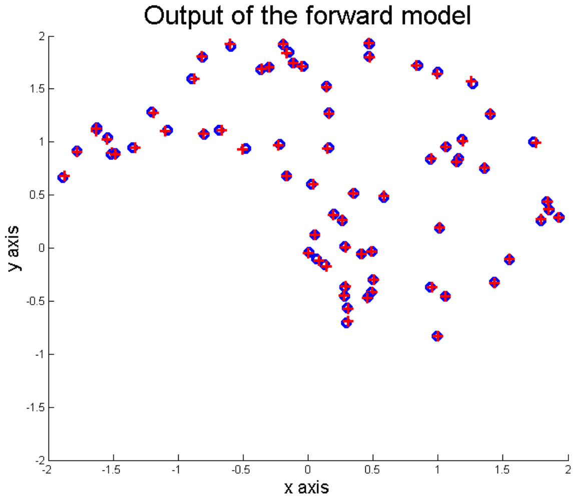


Fig. 6.16. The output of the forward model neural network. Red crosses model the actual end point location of the hand, while blue circles the output of the network. The x, y axes represent the Cartesian coordinates.

The visual observation of the demonstrated movement was also processed by a feedforward neural network. In this second case the network input consists of a noisy version of the demonstrated movement in alocentric coordinates, and was trained to transform it in an egocentric frame of reference. In this context, the noise represents the observer's ability to perceive a behavior correctly. Figure 6.17 demonstrates the output of the training of the neural network using three different noise levels, 0.001, 0.005 and 0.01, respectively.

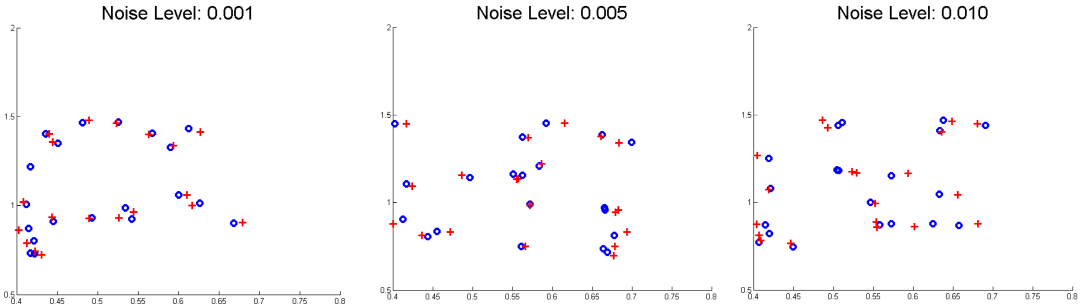


Fig. 6.17. The output of the visual observation model of the agent, using three different noise levels. The x, y axes represent the Cartesian coordinates.

To complete the implementation of the reaching policy the model must learn to derive the distance of the end effector location from the current point in the trajectory. This is accomplished by projecting the output from the forward model and perception pathways in an LSM and using a readout neuron to calculate their subtraction. Experimentally it was estimated that to shape the liquid dynamics and learn this transformation, the dynamic synapses must have delays of approximately 10ms.

Since our model resolution was set to 100ms, we averaged the output of the readout neuron over the 10 steps of the simulation. In Fig. 6.18, we illustrate two sample signals as input to the liquid (top subplot), the output of the readout neuron in the 10ms resolution (middle subplot) and the averaged over the 100ms of simulation time output of the readout neuron (bottom subplot).

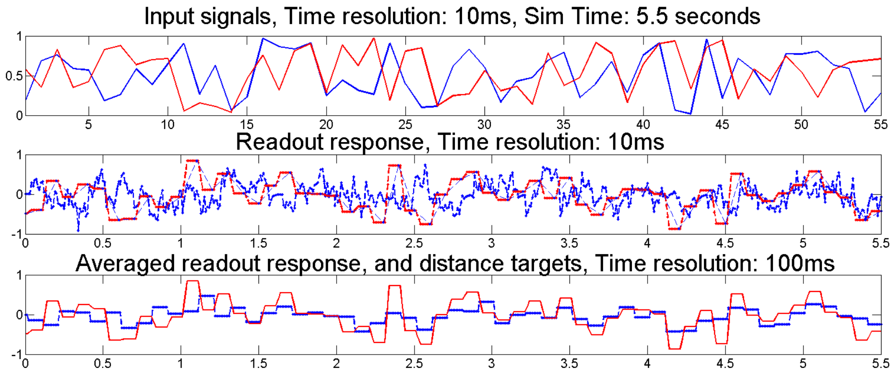


Fig. 6.18. The output of the distance LSM after training. The top plot illustrates two sample input signals of 5.5 seconds duration. The bottom two plots show the output of the neural network readout used to learn the subtraction function from the liquid (middle plot), and how this output is averaged using a 100ms window (bottom plot).

The whole simulation trial lasted 5.5 seconds. As the results show the liquid was able to extract the distance information with a good accuracy. Due to the local control laws used to implement the reaching policy, any small errors in the computation of distance are actually compensated in latter steps.

Online reaching

Having established that the individual pathways/components of the proposed model operate successfully, we now turn our attention to the performance of the model in various reaching tasks. The results presented here are produced by employing the motor control, reward assignment and forward model pathways only.

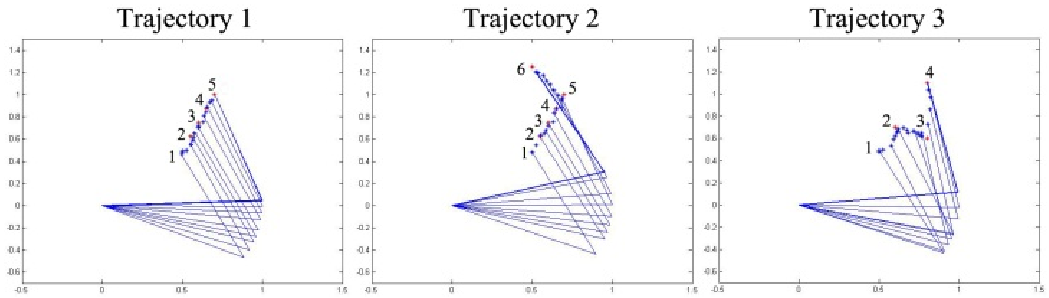


Fig. 6.19. Three trajectories shown to the robot (red points) and the trajectory produced by the robot (blue points). Numbers mark the sequence with which the points were presented.

We note here that the model wasn't trained to perform any of the given reaching tasks, apart from the initial training/imitation period at the beginning of the experiments, shown in Fig. 6.11. After this stage the model was only given a set of points in a trajectory and followed them with very good performance. The first three trajectories we tested were variations of a straight line motion (Fig. 6.19).

As Fig. 6.19 shows the agent was able to follow all three trajectories quite precisely. The average normalized deviation of the agent's position from the points of the trajectory was 0.03 or 3% which shows that the resulting performance was very satisfactory.

In order to evaluate further the performance of the model we used two more complex trajectories. The first required the robot to reach towards various random locations spread in the robot's workspace (Fig. 6.20, Trajectory 1) while the second complex trajectory required the robot to perform a circular motion in a cone shaped trajectory (Fig. 6.20, Trajectory 2). Figure 6.20 illustrates how the aforementioned trajectories were followed by the robot.

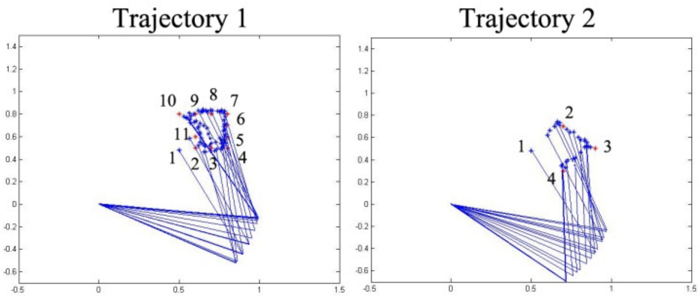


Fig. 6.20. Two complex trajectories shown to the robot (red points) and the trajectories produced by the robot (blue points). Numbers mark the sequence with which the points were presented.

To evaluate the performance of the model quantitatively we created 100 random trajectories and tested whether the agent was able to follow them. Each of these random movements was generated by first creating a straight line trajectory (Fig. 6.21, left plot) and then randomizing the location of 2, 3 or 4 of its points; an example is illustrated in Fig. 6.21, right plot. The error was calculated by summing the overall deviation of the agent's movement from the points in the trajectory for all the entries in the dataset. The results indicate that the agent was able to follow all trajectories with an average error of 2%. This suggests that the motor control component can confront, with high accuracy, any reaching task.

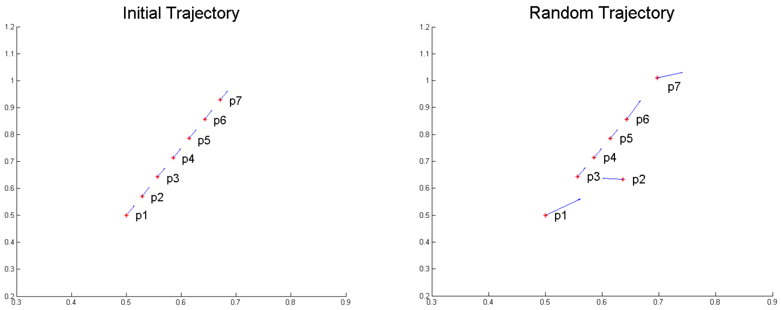


Fig. 6.21. The template used to generate the random test set of 100 trajectories (left plot) and a random trajectory generated from this template (right plot).

State estimation

The role of the self-organizing map in the state estimation pathway was to discretize the end point positions of the observed or executed movements into symbolic labels. The map was trained during the execution phase, where the agent was taught to perform elementary reaching behaviors. The capacity of a network, i.e. the number of nodes it uses to form the map, directly affects the detail of the representation as illustrated in Fig. 6.22. A map with 144 labels (Fig. 6.22, 3rd subplot) can represent the output space of a behavior more accurately. However, as we will demonstrate later in this section, a very detailed map can cause problems during the observational learning phase where the observation class labels must coincide with the execution ones.

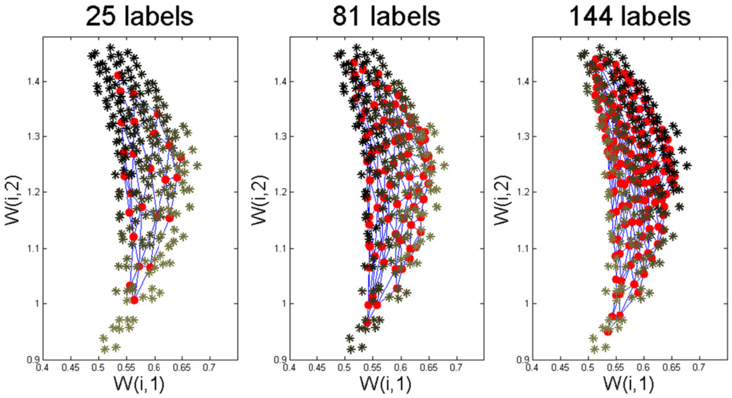


Fig. 6.22. Output of the training of three SOM maps with different capacities for labels. In the first case the map consisted of 25 labels, in the second of 81 labels and in the third of 144 labels. Red circles illustrate the symbolic labels of the map, while the black crosses the training positions of the agent's movements.

In the current simulations we have created three maps of different sizes, 25, 81 and 144 labels respectively (see Fig. 6.22). All three maps were evaluated during the observational learning stage against their ability to produce the same labels during execution and observation.

Visual observation pathway

For the visual input of the agent we have used an LSM to classify three different objects. As discussed in section 6.3, the iconic projection of the object was first filtered and then converted to neural code, where an LSM was trained to classify it based on 3 labels. The following figure presents the classification from the linear classification readout that was trained for this task.

As Fig. 6.23 illustrates, the readout was able to produce stable responses for all three objects irrespectively of the transient states of the liquid. In all three cases the mean average error (mae) was below 0.1, while the mean squared error (mse) had an average value of 0.1. This suggests that the LSM was able to classify the three objects with very high accuracy.

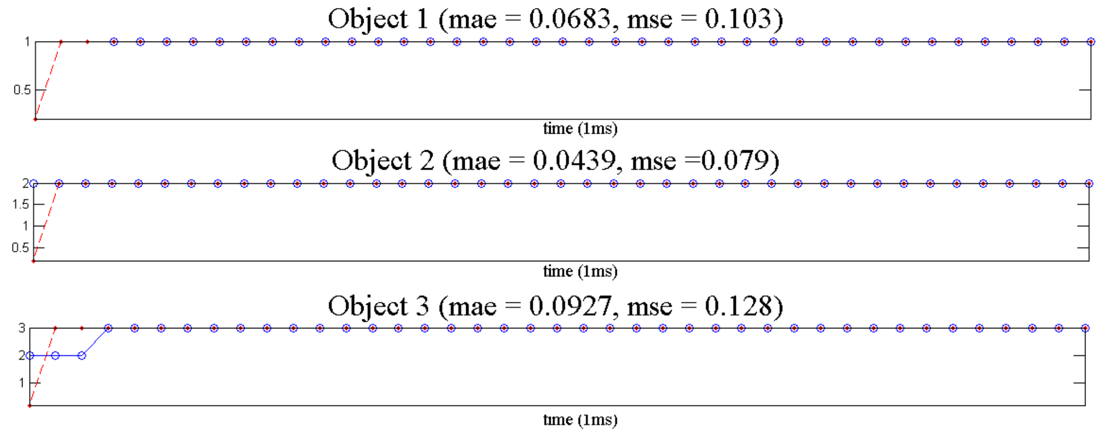


Fig. 6.23. The output from the linear classification readout that was used to classify the three objects. The y axis represents the output class label while the x axis the time step in the simulation.

Higher-order motor control circuit and attractor tuning

As mentioned in the previous section, the higher-order motor control circuit consists of 2 liquids which model the descending force that is exerted to the hand of the agent. Herewith we present the results of the attractor liquid, i.e. the sub-circuit that was developed in order to model the dynamics of a continuous attractor. The following figure illustrates the spike after potentials of the liquid circuit that was described in section 6.3, for 1000ms simulation, as well as the response of the attractor during this period. As Fig. 6.24 illustrates, despite the varying liquid response, the attractor is able to produce a stable response.

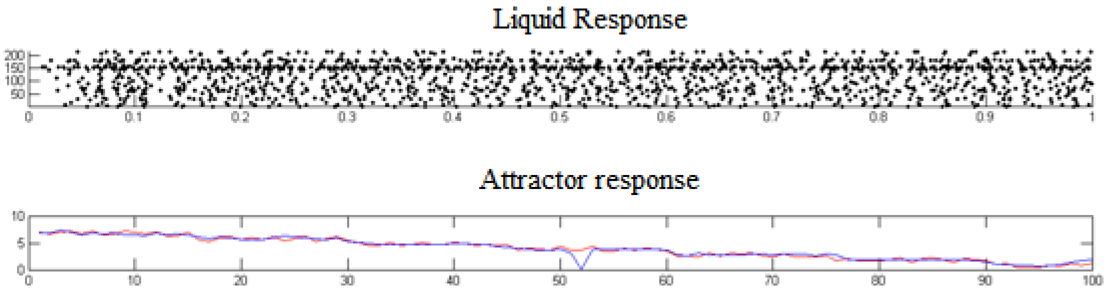


Fig. 6.24. Output of the training of the attractor circuit. The top plot illustrates the response of the liquid while the bottom one the output of the readout that is used to simulate the attractor. The x axis in all plots represents time in 100ms intervals.

After the tuning of the attractor, the circuit was disconnected from the simulated input and was able, given any input, to replicate the attractor dynamics required. Using this optimized liquid, we then implemented a linear regression readout that was trained to model the exact force that is exerted to the hand during reaching. This was accomplished by designing the trained attractor to project directly to the second liquid column of the higher-order motor control pathway, and using one linear regression readout to output the force that must be exerted during the execution of the behavior. Figure 6.25

illustrates the output of the readout unit, which can produce a stable response despite the varying liquid dynamics. The output of the readout is used to inhibit directly the force produced by the motor control component. As the figure shows, the attractor dynamics cause the output of the readout to descent to zero after the first steps of the simulation.

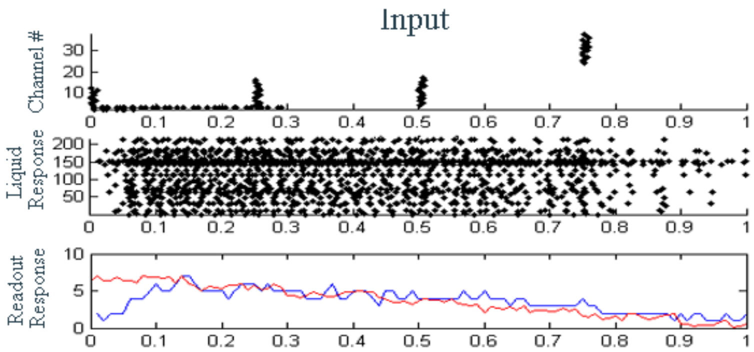


Fig. 6.25. The output of the trained readout that models the force exerted by the higher-order motor control pathway (bottom subplot, blue line) and the desired force value (bottom subplot, red line) for the same period. The top subplot illustrates the input to the circuit, while the middle subplot the liquid dynamics. The x axis in all plots represents time in 100ms intervals.

Observational learning

In this section we illustrate the results from the observational learning experiments, which involve the function of all the pathways of the model. To test the ability of the agent to learn during observation we have generated sample trajectories by simulating the model using predefined parameters. The agent was demonstrated one trajectory at a time, and its individual pathways were trained for 1500ms. The role of this phase was to evaluate the agent's ability to learn a demonstrated trajectory only by observation, i.e. without being allowed to move its hand. Subsequently, to evaluate the extent to which the agent learned the demonstrated movement we run an execution trial, where the agent was required to replicate the demonstrated behavior.

In the following figure we illustrate one sample trajectory that was demonstrated to the agent (Fig. 6.26, left subplot, red circles), the output of the higher-order motor control module (Fig. 6.26, bottom right subplot), the corresponding class labels generated by the model and the mental imagined state of the agent during a 1500ms trial (Fig. 6.26, top right subplot). The noise used for the visual observation pathway was 0.05 (i.e. on each sample a value was added drawn from a Gaussian distribution with mean 0.05), while the size of the state estimation map was 81 labels. As the figure illustrates (Fig. 6.26, left subplot) the agent was able to keep track of the demonstrated trajectory with very good accuracy. The top right subplot in Fig. 6.26 illustrates the class labels that were generated by the state estimation pathway using the visual observation pathway's output in green and the states (in red) that would have been generated if the agent was executing the behavior covertly.

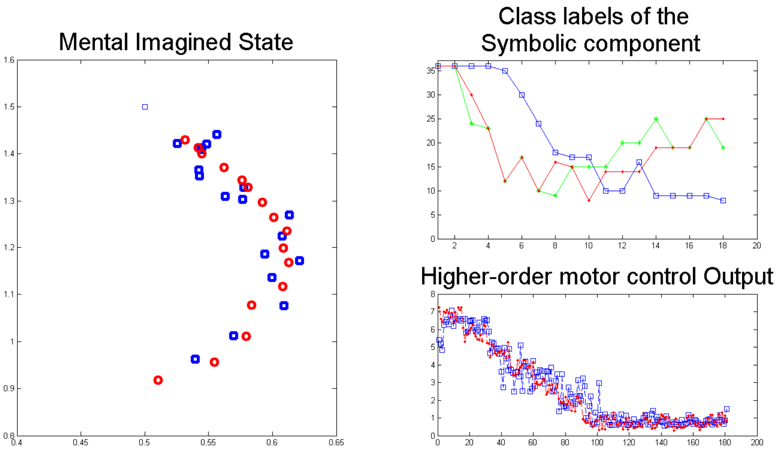


Fig. 6.26. The mental imagined estimate of the hand during observational learning. The left subplot illustrates the trajectory demonstrated (red circles) to the agent and the trajectory imagined by the computational model (blue stars). The top right subplot illustrates the output of the SOM in the state estimation pathway while the bottom right subplot illustrates the output of the linear regression readout in the higher-order motor control pathway (red circles are the desired values of the force while blue boxes are the output of the readout).

As the plot illustrates the two lines coincide to a large extent. To verify that the agent learned the new behavior after the observational learning phase we run the same simulation using the forward model for the state estimation. The following figure illustrates the trajectory that was actually executed by the agent during execution.

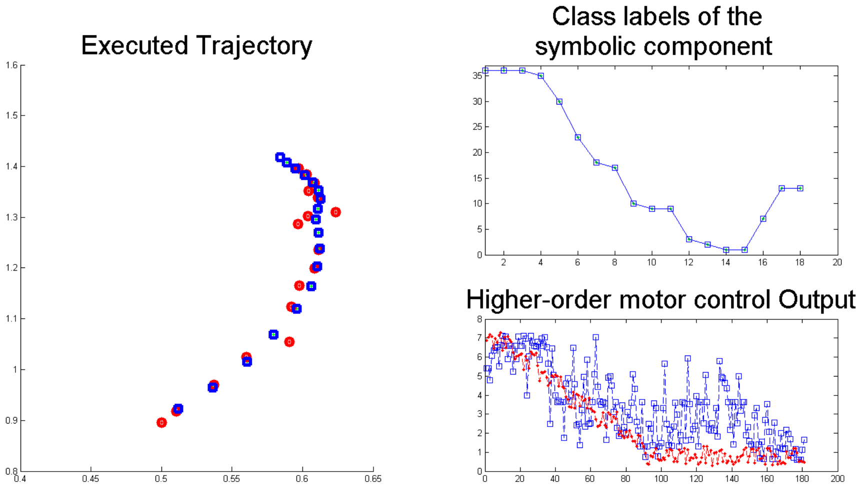


Fig. 6.27. The trajectory executed by the robot during the execution phase. The left subplot illustrates the trajectory demonstrated (red circles) to the agent and the trajectory imagined by the computational model (blue stars). The top right subplot illustrates the output of the SOM in the state estimation pathway while the bottom right subplot illustrates the output of the linear regression readout in the higher-order motor control pathway (red circles are the desired values of the force while blue boxes are the output of the readout).

As discussed in section 6.3, the noise levels and size of the state estimation map had a direct effect on the performance of the model. Larger noise levels altered the perception of the agent and compromised its ability to mentally keep track of the observed movement. Figure 6.28 illustrates the response of the agent for 0.1 and 0.2 noise levels.

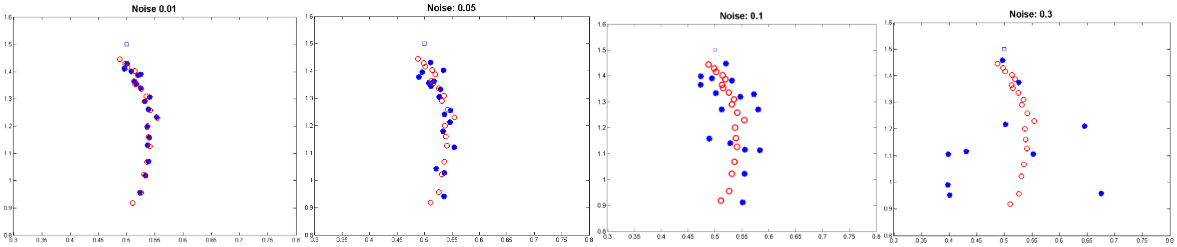


Fig 6.28: The output of the observation pathway under the influence of 4 different noise levels. As the four plots illustrate, the model's perception capabilities are compromised by the existence of noise.

As the results of this section demonstrate the developed computational agent is able to acquire new motor skills only by observation. The quality of learning is correlated with the agent's perceptual abilities, i.e. the extent to which it can perceive an observed action correctly. This skill is facilitated due to the design of the model that allows new motor skills to be learned based on a set of simple parameters that can be derived only by observation.

Chapter 7

Discussion

In the current chapter we discuss the cognitive and computational implications of the two neurophysiological models we have presented throughout the thesis. For this reason we focus on issues regarding model complexity, and whether the modeling resolution used was able to capture the underpinnings of the biological systems that were modeled. In addition, we consider contemporary theories, regarding model development and control, and attempt to explain the emergent properties exhibited by both models.

The discussion is structured in four sections. In the first (section 7.1), we revisit generic contributions on the computational issues that underpin the Liquid State Machines. We then continue to analyze the properties of the first computational agent (section 7.2), focusing on how they relate to the neurophysiological model of Macaques during observation. In the third section (7.3), we overview the implications from the human model, and discuss its most important properties. The chapter is concluded by suggesting future work, and the issues that we plan to focus based on the development of the two models (section 7.4).

7.1 Measuring the separation of Liquid State Machines

One of the main contributions of this thesis was the development of a computational method for improving the classification capabilities of LSMs, by focusing on the separation property of the liquid. This was calculated using the Fisher's Discriminant Ratio, a measure that is maximized when the class means are far from each other, and the class variances are as small as possible. To evaluate the FDR measure against a broad range of classification tasks, we incorporated different types of neuron encodings for the input. As the results show, the FDR criterion accurately predicts the performance of the readouts without having any knowledge on the algorithm used to train them. Due to this fact, the GA, by minimizing the value of the FDR criterion, also improved the performance of all considered classifiers.

In contrast to other criterions, the FDR is a supervised measure, i.e. requires the class labels in order to compute the quantities in eqs. (4.6-4.9). Consequently, the evolutionary framework that was presented is also supervised. We consider this a benefit of the method, since it allows the design of liquid architectures that are suited to the specific dynamics of a general classification task.

Moreover, one important issue about the FDR measure is that it does not make any assumptions about the structure of the data. For instance, other methods of class separability, such as the divergence or Bhattacharyya distance must make a Gaussian assumption in order to be computed. In contrast, the FDR measure is constructed from simple low-level criteria, that describe the geometrical scatter of feature vectors in the problem space, and therefore does not make any assumption about the input data.

This work has a potential to be expanded in order to investigate new ways of reducing the computational load of LSMs, by exploiting additional methods for increasing the separation between different classes. In this context, we will evaluate whether transforming the liquid states, by projecting them along the eigenvectors of the argument of the FDR measure, can increase their separation. This addition can complement the proposed methodology and offer a concrete set of mathematical tools that can evaluate and further advance the computational performance of LSMs. Moreover, the application of the proposed framework in relevant tasks seems very promising. Candidate tasks are the ones that can readily benefit from the enhanced computational abilities of the proposed LSM structure, such as (a) the development of large-scale computational models that are composed of multiple components with distinct functionalities and (b) scene segmentation in video sequences.

7.2 Cognitive implications of the Macaque model

Recent neuroscientific experiments investigated the activation of regions in the brain cortex of primates when observing or executing a grasp behavior (Raos et al., 2007; Raos et al; 2004; Evangeliou et al., 2008; Kilintari et al., 2010). Results from these studies indicate that the same pathway of regions was activated during observation and execution of a behavior. This has led researchers to believe that during observation the primate is internally simulating the observed act, using its own execution circuits to comprehend it.

The above mentioned neuroscientific studies suggest that the mirror neurons are only part of a larger circuitry that is employed by primates in order to simulate an observed behavior (Savaki, 2010). This circuitry involves regions in the primary somatosensory and motor cortex, as well as regions in the parietal lobe (Evangeliou et al., 2008). The current thesis, based on the neuroscientific results of (Raos et al., 2007; Raos et al; 2004), suggests how this extended overlapping pathway can be reproduced in a computational model of action observation/execution, and used for implementing learning by mere observation. The model has allowed us to derive informative predictions regarding the possible reasons for the overlapping activations during observation, as well as the role of specific regions in the circuitry. Given the limitations of a computational model to describe the cortical functions of the brain, these predictions are only considered and discussed in the context of computational modeling.

The information processing properties that characterize the regions involved in the execution and observation of grasping tasks in primates were modeled based on three computational principles. The binding of neural representations across different networks was modeled using associative learning synapses implementing the STDP rule (Song et al., 2000). The feedback required for teaching the agent new behaviors is modeled using a derivation of the Bienenstock-Cooper-Munro learning rule for spike neurons (Baras and Meir, 2007). This algorithm only requires a reinforcement learning signal (which in our case is binary), instead of a supervised one. Finally, following the assumption that primates already know how to execute a lower primitive behavior well before the experiments, the motor control circuit was configured using a genetic algorithm. This allowed us to generate a computationally adequate network configuration, using higher level fitness functions of the behavioral tasks.

Results from the model evaluation indicate that by shunting the motor output of an observing agent, there are several regions in the computational model that are activated at a lower firing rate, compared to their activation during execution. Computationally, and maybe biologically, the lower activations that those networks exhibit are attributed to the fact that during observation alone the proprioceptive input of the agent is not available. As a result, during observation the number of active afferent projections towards each network is smaller, and therefore its activation is lower.

The average regional activations obtained by our model resemble the ones obtained by neurophysiological experiments. The differences between the model and recorded regional activations are attributed to the fact that for modeling purposes we have considered a subset of the large number of neuronal classes that exist in those regions.

The activations during observation are accomplished using an action correspondence circuitry, within the proprioceptive association pathway, that allows the agent to learn to associate its own joint angles with the joint angles of the demonstrator during the observation/execution phase. As a result, while observing another agent performing a known grasp behavior, the model excites the corresponding neural representations in the parietal regions that refer to its own body posture. The activation of those regions during observation is, at least computationally, responsible for partially activating the remaining regions in the model. In the current implementation, this correspondence problem is only solved in action space, i.e. the agent matches its own actions with the actions performed by the demonstrator (Dautenhahn and Nehaniv, 2002). This is accomplished in the STS-VIP-SPL circuitry, where the joint values of the demonstrator from STS are translated to a distributed code in VIP (representing how the agent perceives the observer's action) and matched with the agent's own action, coded in SPL. Due to the existence of SPL in this pathway, the agent must also learn to manipulate its own body before relating its effectors to the demonstrator's. More elaborate versions of this correspondence association mechanism could include mapping between dissimilar bodies, mapping of the state space of the demonstrator or even associating body postures perceived in different frames of reference (Dautenhahn and Nehaniv, 2002).

In addition to the overlapping pathway of regions activated during observation and execution, our model exhibits another important property. It shows that, not only the same regions, but also the same neural representations are activated during observation, a fact that as we have shown facilitates learning. This effect is attributed to the STDP synapses and the linear (outside the refractory period) integration (eq. 4.5) that is used to model the incoming current of the neuron. The importance of these two properties becomes more evident when looking at how neural representations are formed during training of the model. During the observation/execution phase, the synapses that use STDP undergo a competition process for the control of post-synaptic cell firings (Song et al., 2000). The synapses that lose the competition depolarize, leaving only those that represent a causal relationship between two neurons to survive. The term causal here is used to describe the relationship between the response behavior of the pre- and post-synaptic neurons. In contrast to Hebbian learning, that treats causality as correlation (derived from cell co-activation), STDP requires a pre-synaptic neuron to fire persistently before the post-synaptic action potential in order to strengthen the connection. As a result, in the current model implementation, after the observation/execution phase, only the synapses that describe a robust association (i.e. persistent and mutually correlated at the level of individual spikes) between their two neurons would remain strengthened. Consequently, during the observational learning and observation alone phases the networks continue to activate the same neural representations as before even though some of the inputs to the model are kept silent.

The fact that these common neural representations continue to be activated, and further strengthened, even when neurons fire with lower frequencies, is attributed to the linear model used for integrating pre-synaptic input (eq. 4.5). More specifically, even though all networks in the model exhibit a lower firing frequency during observation and observational learning phases, the action potentials of their neurons maintain their temporal relationship. Thus, neurons that are active in these common representations will exhibit proportional firing frequencies (i.e. a presynaptic neuron that fires just before a post-synaptic neuron during the observation/execution phase, will continue to do so during the observation phase, because both neurons fire with lower frequency). We also point out that the retrieval of a pattern using partial input can be accomplished using Hebbian synapses (Billard and Hayes, 1999; Chaminade et al., 2008). However given the architecture of the model networks and the choice of the neuron model of the current implementation, the use of STDP connections allows to better address certain issues that would require additional considerations if Hebbian synapses were used.

To develop the properties of the neurons in the $F5_{mirror}$ network we have used two computational principles, associative learning and input normalization. In computational neuroscience, theories regarding the formation of mirror neurons fall into two categories. The first, adaptation hypothesis, is implicitly suggested in various articles (Rizzolatti and Craighero, 2004; Ramachandran, 2000), and claims that mirror neurons are the result of ontogenetic development, and thus most of their wiring is genetically inherited. The theory is supported by experimental evidence that demonstrates the ability of infant primates to imitate (Meltzoff and Decety, 2003; Ferrari et al., 2003; Ferrari et al., 2006). According to the adaptation hypothesis, mirror neurons have been favored by natural selection due to their properties that facilitate action understanding, an important component of social cognition. The second theory suggests that mirror neurons are the result of associative learning following the experience of the primate with grasping execution and observation (Heyes and Ray, 2000; Heyes, 2001). The theory attributes the formation of mirror neurons to common sensorimotor contingencies, i.e. correlated experience caused by simultaneously observing and executing an action. To explain the transitive property of mirror neurons, the associative learning hypothesis has used concepts such as stimulus generalization, i.e. the modulation of a neuron's response to a stimulus, caused by differences between the learning stimuli and the experienced stimuli (Pearce, 1987).

More recently, new hypotheses on the formation of mirror neurons are starting to emerge, that suggest only a certain amount of genetic predisposition at the neuron level in mirror neurons (Lepage and Theoret, 2007). Attempts to combine the two theories together have also been made, which show that mirror neurons could have been the result of ontogenetic development and associative learning (e.g. Del Giudice and Manera, 2009).

The approach used in this model is consistent with the associative learning hypothesis. More specifically, in the $F5_{mirror}$ network of the first agent, the neurons acquire their matching properties during the first observation/execution learning stage, while the simulated agent observes and learns to execute an action simultaneously. The associative STDP synapses and the correlated activity caused from the observation and execution streams being active at the same time facilitates their development. The normalization constants are used to fit into the model the tendency of mirror neurons to fire only when an object is present, i.e. when goal-directed actions are involved. Their use allows the $F5_{mirror}$ network of the model to shape its response in respect to the aforementioned property, even though only some aspects of their cortical connections are considered.

In the Macaque brain, populations of mirror neurons have been found in the ventral premotor area F5 (Gallese et al., 1996) and the inferior parietal lobe (Gallese et al., 2002). For the human EOMS, even though direct cell recordings are not available, fMRI and PET imaging data show that it expands to areas beyond those two regions (Caspers et al., 2010). However there is still an ongoing debate on whether these mirror neurons are part of a common or an overlapping neural code. For example a recent fMRI study in human aIPS (Dinstein, 2008) has showed that even though there are some overlapping parts of a neural code during executed and observed movements, their spatial pattern of activities is distinctly different. Both the overlapping and common code interpretations of the neuronal activations during execution and observation have their pros and cons. If the neural codes shared during observation and execution are common, then building the pathway of networks for observation should only be concerned with replicating that of execution. However in this case additional issues such as resolving the sense of agency should be dealt at the level of inter-regional activations of these regions. If, on the other hand, the neural representations are only overlapping during execution and observation, then the non-overlapping parts of the representation could be used to deal with issues such as agency attribution.

The model also makes an informative prediction, regarding the role of the superior parietal cortex, and in particular region SPL in the design of imitation models. Even though this area is usually neglected by the computational modeling community, our results show that it could be used for implementing an action correspondence mechanism in artificial agents. Cortically, the region contains neurons that are somatotopically organized to respond to various joint and skin stimulation of an agent's body, and for this reason SPL has been attributed with holding a representation of the body schema of the agent (Sakata et al, 1973). The recently found activation of SPL during observation of grasping behaviors, reported in the results of (Evangeliou et al., 2008), opens a whole new perspective on the possible role of the region during imitation tasks. In the context of computational modeling, given the activation of this region during observation alone, the strong projections it has with regions SI and the connections from IPL, the model shows that SPL can activate accordingly in order to respond to the actions of the agent and the demonstrator.

7.3 Cognitive implications of the Human model

The final issue that we discuss is the computational implementation of observational learning based on human neurophysiological data. For this model, we have exploited the fact that when observing, human primates activate the same pathway of regions as when executing. The cognitive interpretation of this fact is that when we observe, we use our motor system, i.e. our own grounded motor experiences, in order to understand what we observe. The developed model was able to learn new motor skills by mere observation, by adopting the peripheral, higher-order motor control, component of its motor system during observation.

The fact that the brain activates the same pathways to simulate an observed action is an important component of human intelligence, and as it has been suggested a basis for social cognition (Baron-Cohen et al., 1993; Saxe et al., 2004).. In the computational modeling community, most research in this area has focused on the function of mirror neurons. The evidence of activating pathways throughout the cerebral cortex suggests that the cortical overlap of the regions is much more extended than the mirror neuron areas, in agreement with the mental simulation theory. More importantly, since action observation activates the same regions as in action execution, motor imagery and observational learning can be used to revise our understanding about the content of motor representations. Computational models such as the one presented can help us understand the basis under which all these processes can work together in order to accomplish a behavioral task.

Modularity has been an important aspect of the modeling process. At the cortical level modularity was implemented using pathways of interacting regions that realize a specific cognitive function. At the motor level, modularity was applied by designing motor synergies, i.e. pre-organized motor circuits that coordinate several muscles of the agent’s body. This allowed us to decompose the complex process of motor control and implement all cognitive functions separately. The final output of the model was then implemented as a coordinated process of all the underlying sub-systems.

The final model was designed to perform two main functions: (i) online reaching, i.e. enable the agent to reach towards any given location with very little training, and (ii) observational learning, i.e. the acquisition of novel skills only by observation. To implement the reaching component we have devised a local reaching policy, where the computational model exerts forces that move the hand towards a desired location. The benefit of this approach is that any errors in the movement can be compensated at the latter stages of motor control. To implement observational learning of reaching we first examined the content of the motor representations in the cerebral cortex of human and monkey primates. For this reason we have structured the motor control component into an innate reaching control system and a peripheral higher-order motor control system. Learning during observation was implemented on the higher-order motor control component based on simple control parameters. This intuition was very important for the implementation of the observational learning process, since the agent wasn't required to restructure its whole motor system in order to acquire new behaviors.

7.4 Future work

In the current thesis we have developed two computational models of observational learning, inspired by the neurophysiology of human and Macaque primates. In both cases we have made use of the fact that primates during observation activate a pathway of regions that overlaps extensively with the one developed for action execution. Due to this, both agents were able to activate their own motor representations in order to acquire new knowledge.

Having established a working model of observational learning, one of the important aspects that is interesting to investigate in the future is the cortical underpinnings of motor inhibition during observation. More specifically, what are the reasons that cause the human's body to stay immobile when only observing others. Cortical inhibition must exist at the spinal levels by preventing the excitation of the muscle reflexes. In this context, it is interesting to exploit possible implementations of the cerebellum, and how its function can facilitate the inhibition of specific components of the movement.

Moreover, the developed models can provide a basis for implementing agency attribution, i.e. the process that allows the cortical agents to perceive their body as their own. Despite its potential benefits, agency attribution has received little attention within the computational modeling community. Early attempts to investigate the process focused on self-recognition, by employing purely visual (Berthouze and Itakura, 1997; Michel et al., 2004) or pre-coded modules (Billard and Mataric, 2004), in order to derive whether an agent is the initiator of its own movement. These approaches however do not consider the structure and function of the motor control system which, as suggested by cognitive neuroscientists, has a pivotal role in the perception of an action (Jeannerod, 2003).

More recent models facilitate agency attribution by exploiting the incongruence between the sensorimotor representations of an acting and an observing robot. In (Tani, 1998), the authors implement a robot that can identify its own actions, using a comparison module that estimates whether the motor commands and body state of the agent coincide. A similar intuition is used in the models of (Wolpert and Kawato, 1998; Haruno et al., 2001), which resolve the function by monitoring the causal relationship between an action and its afferent sensorimotor cues. In cases where a predicted action effect, derived using forward models, coincides with the motor control commands the behavior is attributed to the acting agent (Demiris and Meltzoff, 2008).

There are several problems associated with the above approaches. They don't make a clear distinction between the sense of agency and sense of body ownership, which as suggested by cognitive neuroscientists pertain to different processes (Synofzik et al., 2008). As a result, the function of assigning the correct agent to an action is reduced to a matching process, between the body state and the motor output, a computation that is not always guaranteed due to the noise and non-linearities inherent in motor control (Synofzik et al., 2008).

Another common property of the aforementioned methods is that they use modal architectures that generate a static and momentary distinction between the observed and executed behaviors. This approach neglects the large number of representational levels that exist in the motor image of an acting agent (Jeannerod, 2003), and could be exploited in order to derive a more robust implementation of agency attribution. More importantly, due to their commitment to action execution (a comparison is only possible if the agent is acting), they cannot be applied to a large category of sub-processes of imitation, including the mental simulation models discussed in this thesis, where the results of the action are covert and unavailable to the agent (Hourdakis et al., 2011; Hourdakis and Trahanias, 2011b).

To confront these issues one can employ the motor control system of the two computational agents, in order to investigate how the functional activations in the overlapping pathway of regions can be used to distributively resolve the agency of an observed movement. This consists of a novel method for confronting the problem, by employing the underpinnings of the robot's motor control system, instead of its momentary output, in order to identify the correct agent of an action.

This new approach has important advantages. Computationally, distributed representations are more robust and can generalize better to unknown domains. Moreover, by employing the observer's motor image, the method will utilize all the subliminary processes that participate in the recognition of an action in order to resolve its agency. This involves a large number of cognitive functions, including attention, action planning, goal identification and motor execution. Consequently, the robots will be able to develop a conceptual representation of agency that is based on the same spatio-temporal properties that characterize their motor control system.

Another important aspect that one can consider for future work is the enrichment of the computational modeling methodology presented in chapter 4, in order to be able to consider quantitative information derived directly from neuroscientific data. In this context, it is important to investigate how low level data, such as the neuronal tuning curves, and higher level regional activations can be integrated within the same computational context. To accomplish this we plan to develop a novel neural network that will provide appropriate structures in order to achieve this purpose. It will consist of generic biologically inspired neurons, and provide a convenient framework for the computational modeler to customize its neurons, in order to exhibit the appropriate responses at the cellular level, as well as methods for combining the output of neuronal ensembles in order to accomplish certain behavioral functions. Moreover, we plan to consider how this network will be combined with the pathway computational modeling methodology, in order to facilitate the development of large scale, biologically inspired computational models.

All in all, the work presented in this thesis has the potential to lead to novel methods for implementing mental simulation, a core cognitive function in primates that has long underpinned their social interactions. By taking advantage of the methods described in this thesis, computational modelers have the potential to develop artificial agents that will build and develop their cognitive system without interacting physically with their environment. As a result, robots can engage into an ongoing and continuous learning process, in which even though they do not consume any excessive power load, they continue to learn from their social partners, in a similar manner as humans do. Moreover, by combining the perspective of developing methods for resolving the sense of agency, robots will be able to construct a mental image of themselves, which will be grounded in their sensorimotor interactions with the world. Consequently, they will be able to develop intersubjective representations, that will facilitate their social interactions, such as empathy, language, emotions and perhaps even consciousness.

Bibliography

Abbott, L. F. and Kepler, T. B. (1990). Model neurons: from Hodgkin Huxley to Hopfield. *In Statistical Mechanics of Neural Networks*, ed. by Garrido, L., Lecture Notes in Physics, v.368, pp.5-18, Springer.

Abeles M., (1991). Corticonics: Neural Circuits of the Cerebral Cortex. Cambridge University Press, Cambridge.

Adrian, E.D., (1926). The impulses produced by sensory nerve endings, *The journal of physiology*, v.61:1, pp.49-72, Physiological Society.

Alexander R.M., (1997). A minimum energy cost hypothesis for human arm trajectories. *Biol Cybern,* v.76, pp.97–105.

Alissandrakis, A., Nehaniv, C. L. and Dautenhahn, K., (2002). Imitating with ALICE: learning to imitate corresponding actions across dissimilar embodiments, *IEEE Transact. Systems, Man & Cybernetics, Part A: Systems Hum*. v.32, pp.482–496.

Amari, S. and Takeuchi, A. (1978). Mathematical theory on formation of category detecting nerve cells. *Biological Cybernetics*, v.29, pp.127–136.

Amit, D. and Brunel N., (1997). Model of global spontaneous activity and local structured activity during delay periods in the cerebral cortex, *Cerebral Cortex,* v.7:3, pp.237-252.

Amit D.J. (1989). Modelling brain function: the world of attractor neural networks, Cambridge University Press, Cambridge, England.

Anderson, M.L. (2003). Embodied cognition: A field guide, Artificial Intelligence, v.149:1, pp.91-130.

Arbib M.A. (2004). From Monkey-like Action Recognition to Human Language: An Evolutionary Framework for Neurolinguistics, *Behavioral and Brain Sciences*, v.28, pp.105-167.

Arbib, M.A.,Iberall, T. and Lyons, D. (1990). Schemas that integrate vision and touch for hand control, *Vision, brain, and cooperative computation*, MIT Press.

Arbib, M. A. and T. Iberall (1990). Schemas for the control of hand movements: An essay on cortical localization. *The control of grasping. (eds). M. Goodale. Ablex Norwood*: pp.163-180.

Arbib, M., (1981). Perceptual structures and distributed motor control. *Handbook of physiology*, v.2, pp.1449-1480.

Arbib, M.A. (1992). Schema Theory, Encyclopedia of artificial intelligence, v.2, pp.1427-1443, Wiley New York.

Arbib, M. A., Billard, A., Iacoboni, M. and Oztop, E. (2000). Synthetic brain imaging: grasping, mirror neurons and imitation. *Neural Networks,* v.13, pp.975–997.

Arbib, M.A. (2003). The handbook of brain theory and neural networks, The MIT Press.

Ariff G.D., Donchin O., Nanayakkara T., Shadmehr R. (2002). A real-time state predictor in motor control: study of saccadic eye movements during unseen reaching movements. *J. Neurosci*. v.22, pp.7721–7729.

Arkin, R.C. (1998). Behavior-based robotics, The MIT Press.

Arulampalam, S., Maskell, S., Gordon, N. and Clapp, T. (2002). A tutorial on particle filters for on-line non-linear/non-Gaussian Bayesian tracking. *IEEE Trans. Signal Processing,* v.50, pp.174–188.

Astafiev S.V., Stanley C.M., Shulman G.L., Corbetta M., (2004). Extrastriate body area in human occipital cortex responds to the performance of motor actions. *Nat Neurosci,* v.7, pp.542–548.

Atkeson C. G. and Schaal S., (1997). Learning tasks from a single demonstration. *In IEEE Int. Conf. Robotics Automation (ICRA97)*, v.2, pp.1706–1712. Piscataway, NJ: IEEE.

Atkeson C.G. (1989). Learning arm kinematics and dynamics. *Annual Review of Neuroscience*, v.12, pp.157-183.

Auer P., Burgsteiner H., Maass W. (2008). A learning rule for very simple universal approximators consisting of a single layer, Neural Nets, v.21, pp.786-795.

Bagnell J., Kadade S. Ng. and Schneider J. (2004). Policy search by dynamic programming. *In Advances in Neural Information Processing Systems 16 (NIPS)*, Vancouver, BC, CA.

Bair, W. and Koch, C. (1996). Temporal precision of spike trains in extrastriate cortex of the behaving macaque monkey, *Neural computation,* v.8:6, pp.1185-1202.

Ballard, D.H. (1986). Cortical connections and parallel processing: Structure and function. *The Behavioral and Brain Sciences*, v.9, pp.67-120.

Baldissera F., Cavallari P., Fournier E., Pierrot-Deseilligny E. and Shindo M. (1987). Evidence for mutual inhibition of opposite Ia nterneurones in the human upper limb. *Exp. Brain Res*., v.66, pp.106–114.

Baldissera F, Cavallari P, Craighero L, FadigaL. (2001). Modulation of spinal excitability during observation of hand actions in humans. *Eur. J. Neurosci*. v.13, pp.190–94.

Bandura, A. and Wood, R. (1989). Human agency in social cognitive theory. *Am. Psychol*. v.44:9, pp.1175–1184.

Baras, D. and Meir, R. (2007). Reinforcement learning, spike-time-dependent plasticity and the BCM rule. *Neural Computation,* v.19. pp.2245-2279.

Barbas, H and Pandya, D. N. (1987) Architecture and frontal cortical connections of the premotor cortex (area 6) in the rhesus monkey. *J. Comp. Neurol.* v.256, pp.211–228.

Bard, K. and Russell C., (1999). Evolutionary foundations of imitation: Social, cognitive, and developmental aspects of imitative processes in non-human primates. *Imitation in infancy: Progress and prospects of current research*, p. 89-123.

Barlow, H.B., Kaushal T.P., and Mitchison G. J. (1989). Finding minimum entropy codes. *Neural Computation*, v.1, pp.412–423.

Barlow H.B. (1972). Single units and sensation: a neuron doctrine for perceptual psychology? *Perception,* v1, pp.371-394.

Baron-Cohen, S. (1997). Mindblindness: An essay on autism and theory of mind. The MIT Press.

Baron-Cohen, S., H. Tager-Flusberg, and D. Cohen (1993). *Understanding other minds: Perspectives from autism*. Oxford University Press Oxford.

Barresi, J. and Moore, C. (1995). Intentional relations and social understanding. *Behavioral and Brain Sciences*, v.19, pp.107–154.

Barsalou L. and Wiemer-Hastings K. (2005). Situating abstract concepts. *Grounding cognition: The role of perception and action in memory, language, and thinking*, pp.129-163.

Barto A. G., Sutton R. S. and Anderson C. W. (1983). Neuronlike adaptive elements that can solve difficult learning control problems. *IEEE Transactions on Systems Man and Cybernetics*, v.13, pp.834–846.

Barto, A.G. (1995). *Adaptive critics and the basal ganglia, Models of information processing in the basal ganglia*, MIT Press, Cambridge.

Barto (1991). Learning and incremental dynamic-programming, *Behavioral and Brain Sciences*, v.14:1, pp.94.

Bellman, R. (1957). Dynamic Programming, Princeton Univ. Press, Princeton, New Jersey.

Billard, A. (2000) Learning motor skills by imitation: a biologically inspired robotic model. *Cybern. Systems,* v.32, pp.155–193.

Billard, A. and Mataric, M. (2001). Learning human arm movements by imitation: evaluation of a biologically-inspired architecture. *Robotics Autonomous Systems,* v.941, pp.1–16.

Billard, A., and Hayes, G. (1999). DRAMA, a connectionist architecture for control and learning in autonomous robots. *Adaptive Behavior*, v.7:1, pp.35–63.

Billard A, Mataric M, (2004) Learning human arm movements by imitation: Evaluation of a biologically inspired connectionist architecture, Robotics and Autonomous Systems, 37(2-3), 145-160.

Binkofski, F., Buccino, G., Posse, S., Seitz, R.J., Rizzolatti, G. and Freund, H.J. (1999). A fronto-parietal circuit for object manipulation in man: evidence from an fMRI study. *European Journal of Neuroscience*, v.11, pp.3276-3286.

Bizzi, E., Accornero, N., Chapple, W. & Hogan, N. (1984). Posture control and trajectory formation during arm mosment. *J. Neurosci.* v.4, pp.2738—2744.

Bizzi E., Mussa-Ivaldi F.A., and Giszter S.F., Computations Underlying the Execution of Movement: A Novel Biological Perspective, Science, v.253, pp. 287–291.

Bizzi E., Cheung VK., d'Avella A., Saltiel P. and Tresch M. (2008) Combining modules for movement. *Brain Res Rev,* v.57:1, pp.125– 133.

Blakemore, C. and Cooper, G. F. (1970). Development of the brain depends on the visual environment. *Nature*, v.228, pp.477–478.

Blakemore, S., Winston J., and Frith U. (2004). Social cognitive neuroscience: where are we heading? *Trends in cognitive sciences*. v.8:5, pp. 216-222.

Boecker H, Dagher A, Ceballos-Baumann, Passingham, R. et al., (1998). Role of the human rostral supplementary motor area and the basal ganglia in motor sequence control: investigations with H2 15O PET. J. Neurophysiol. 79: 1070–1080, 1998.

Bonaiuto, B., Rosta, E., and Arbib, M.A. (2005). Recognizing Invisible Actions, Workshop on Modeling Natural Action Selection, Edinburgh, July 2005.

Borenstein, E., and Ruppin, E. (2005). The evolution of imitation and mirror neurons in adaptive agents. *Cognitive Systems Research*, v.6:3.

Box, H.O. and Gibson, K.R. (1999). Mammalian social learning, Cambridge University Press, Cambridge.

Brass, M. and Heyes, C. (2005). Imitation: is cognitive neuroscience solving the correspondence Problem? *Trends Cogn. Sci.* v.9:10, pp.489–495.

Brass M, Bekkering H and Prinz W. (2001). Movement observation affects movement execution in a simple response task. *Acta Psychol*, v.106, pp.3–22.

Brass, M., Bekkering, H., Wohlschlaeger, A. and Prinz, W. (2000). Compatibility between observed and executed finger movements: comparing symbolic, spatial and imitative cues. *Brain and Cognition*, v.44, pp.124-143.

Breazeal, C. and Scassellati, B. (2002). Robots that imitate humans. *Trends in Cognitive Sciences*. v.6:11, p.481-487.

Brighina F., La Bua V., Oliveri M., Piazza A.. and Fierro B. (2000) Magnetic stimulation study during observation of motor tasks. *J Neurol Sci,* v.174, pp.122–126.

Brodmann, K. (1905). Perceptual Neuroscience: The Cerebral Cortex, Cambridge, MA: Harvard University Press.

Brown T., (1912).The factors in rhythmic activity of the nervous system. *Proc R Soc Lond Ser,* v.85:579, pp.278–289.

Buccino, G., Binkofski, F. and Riggio, L. (2004). The mirror neuron system and action recognition. Brain Lang. v.89, pp.370–376.

Buccino G., Binkofski F., Fink G.R., Fadiga L., Fogassi L., et al. (2001). Action observation activates premotor and parietal areas in a somatotopic manner: an fMRI study. Eur. J. Neurosci. v.13:400, pp. 400-40.

Buccino, G. et al. (2004) Neural circuits underlying imitation learning of hand actions: An event-related fMRI study. *Neuron,* v.42, pp.323–334.

Bugmann, G. (1997). "Biologically plausible neural computation." *BioSystems,* v.40:1-2, pp.11-19.

Buhl E.H. and Halasy K. and Somogyi P. (1994). Diverse sources of hippocampal unitary inhibitory postsynaptic potentials and the number of synaptic release sites, *Nature*, v.368:6474, pp.823-828.

Bullock, D., Contreras-Vidal, J. L. and Grossberg, S. (1992). Equilibria and dynamics of a neural network model for opponent muscle control. In G. A. Bekey and K. Y. Goldberg (Eds.), *Neural Networks in Robotics*, pp.439–457. Norwell, MA: Kluwer.

Bullock, D. and Contreras-Vidal, J. L. (1993). How spinal neural networks reduce discrepancies between motor intention and motor realization. In K. M. Newell and D. M. Corcos (Eds.), *Variability and Motor Control*, pp. 183–221. Champaign, IL: Human Kinetics Publishers

Bullock, D. and Grossberg S. (1988b). Neural dynamics of planned arm movements: Emergent invariants and speed-accuracy properties during trajectory formation. *Psychological Review*, v.95:1, pp. 49-90.

Bullock, D., Grossberg, S. and Guenther, F. H. (1993). A self-organizing neural model of motor equivalent reaching and tool use by a multijoint arm. *J. Cogn. Neurosci*. v.5, pp.408–435.

Bullock D., Grossberg S. (1988a). The VITE model: a neural command circuit for generating arm and articulator trajectories. In: Kelso J, Mandell A, Shlesinger M (eds), *Dynamic patterns in complex systems*. World Scientific, Singapore pp.206–305

Bullock D. and Grossberg S. (1989). VITE and FLETE: neural models for trajectory formation and postural control. *In Hershberger WA (ed) Volitional action*. North-Holland, Amsterdam pp.253– 297.

Burgsteiner, H. Kroll, M., Leopold, A. and Steinbauer, G. (2007). Movement prediction from real-world images using a liquid state machine, Applied Intelligence, v.27:2, pp.99-109.

Bush, P. and Douglas, R. J. (1991). Synchronisation of bursting action potential discharge. *Neural Computation*, v.3, pp.19-30.

Buys E.J., Lemon R.N., Mantel G.W., Muir R.B. (1986). Selective facilitation of different hand muscles by single corticospinal neurones in the conscious monkey. *J. Physiol*. v.381, pp.529–549

Byrne R.W., Russon A.E. (1998). Learning by imitation: a hierarchical approach. *Behav Brain Sci,* v.21, pp.667-712.

Byrne, R. W. (2003). Imitation as behavior parsing. *Phil. Trans. R. Soc. Lond. B,* v.358, pp.529–536.

Byrne, R. W. (1994). The evolution of intelligence. *In Behaviour and evolution* (eds P. J. B. Slater & T. R. Halliday), pp.223–264. Cambridge, UK: Cambridge University Press.

Caminiti, R., Johnson P. and Urbano A. (1990). Making arm movements within different parts of space: dynamic aspects in the primate motor cortex. *Journal of Neuroscience*, v.10:7, pp.2039-2058.

Carr, L., Iacoboni, M., Dubeau, M.-C., Mazziotta, J. C., & Lenzi, G. L. (2003). Neural mechanisms of empathy in humans: A relay from neural systems for imitation to limbic areas. *Proc Natl Acad Sci,* v.100:9, pp.5497–5502.

Carruthers P. and Smith PK (1996). Theories of theories of mind. Cambridge University Press, Cambridge, UK

Caspers, S. and Zilles, K. and Laird, A.R. and Eickhoff, S.B. 2010. "ALE meta-analysis of action observation and imitation in the human brain" v.50 pp. 1148-1167, Neuroimage. Elsevier

Cattaneo L., Fabbri-Destro M., Boria S., Pieraccini C., Monti A., Cossu G., et al. (2007). Impairment of actions chains in autism and its possible role in intention understanding. *Proc Natl Acad Sci*, USA, v.104, pp.17825–17830.

Chaminade T., Meltzoff A. N. and Decety J. (2002). Does the end justify the means? A PET exploration of the mechanisms involved in human imitation. *Neuroimage*, v.15:2, pp.318–328. Chaminade, T. and Oztop, E. and Cheng, G. and Kawato, M. 2008. From self-observation to imitation: Visuomotor association on a robotic hand, Brain research bulletin. v.75:6. pp.775-784. Elsevier

Chaminade, T. and Meltzoff, A.N. and Decety, J. (2005). An fMRI study of imitation: action representation and body schema, Neuropsychologia, v.43:1, pp.115-127, Elsevier

Cheney, P.D., and Fetz E. E. (1980). Functional classes of primate corticomotoneuronal cells and their relation to active force. *J. Neurophysiol*. v.44, pp.773-791.

Chow K.L., and Hutt P.J. (1953). The 'association cortex' of Macaca mulatta, a review of recent contributions to its anatomy and function, *Brain*, v.76, pp.625-677.

Christodoulou C., Bugmann G., Taylor J.G. and Clarkson T. (1992). An extension to the temporal noisy-leaky integrator neuron and its potential applications, *Proc. IJCNN'92,* v.3, pp.165-170, Beijing.

Churchland, P.S. (2002). Brain-wise: Studies in neurophilosophy, The MIT Press.

Cliff D., Husbands P. and Harvey I. (1993). Explorations in evolutionary robotics. *Adaptive Behavior*. v.2:1, pp.73.

Cochin S., Barthelemy C., Roux S., Martineau J. (1999). Observation and execution of movement: similarities demonstrated by quantified electroencephalography. *Eur J Neurosci,* v.11, pp.1839 –1842.

Cohen A., Wallen P. (1980). The neural correlate of locomotion in fish: “fictive swimming” induced in an in vitro preparation of the lamprey spinal cord. *Exp Brain Res,* v.41, pp.11–18.

Cohen, D. (1968). Magnetoencephalography: evidence of magnetic fields produced by alpha-rhythm currents, *Science,* v.161:3843, pp.784-786.

Colebatch J.G., Deiber M.P., Passingham R.E., Friston K.J., Frackowiak R.S.J. (1991).Regional cerebral blood flow during voluntary arm and hand movements in human subjects. *J. Neurophysiol*. v.65, pp.1392–1401.

Collin, C. and Woodburn, R. (1998). Neuromorphism or pragmatism? A formal approach. *In: Neuromorphic systems: Engineering silicon from neurobiology* eds. L. Smith & A. Hamilton. World Scientific

Conditt M.A., Gandolfo F., Mussa-Ivaldi F.A. (1997). The motor system does not learn the dynamics of the arm by rote memorization of past experience. *J Neurophysiol,* v.78, pp.554 –560.

Corbin, C. B. (1967). Effects of mental practice on skill development after controlled practice. *Research Quarterly*, v.38, pp.534-538.

Craighero L., Bello A., Fadiga L. and Rizzolatti G. (2002). Hand action preparation influences the responses to hand pictures. *Neuropsychologia*, v.40, pp.492-502.

Cronin, J. (1987). *Mathematical aspects of Hodgkin-Huxley neural theory*. Cambridge University Press, Cambridge, UK

Crosby, E.C., Humphrey T., and Lauer E.W. (1962). *Correlative Anatomy of the Nervous System*, Macmillan, New York, pp. 449-451.

Crutcher M.D. and Alexander G.E. (1990). Movement-related neuronal activity selectively coding either direction or muscle pattern in three motor areas of the monkey. *J. Neurophysiol*. v.64, pp.151–163.

Csibra, G. (2005). Mirror neurons and action observation. Is simulation involved?, *What do mirror neurons mean?*

Geschwind, N. (1965). "Disconnection syndromes in animals and man." *Brain,* v.88, pp.237-294.

D'Souza A., Vijayakumar S. and Schaal S. (2001). Learning inverse kinematics. *In IEEE Int. Conf. Intell. Robots Systems (IROS 2001)*, Hilton Head Island, SC, 13–15 June 2000.

Damasio A. and Damasio H. (1994). Cortical systems for retrieval of concrete knowledge: The convergence zone framework. *Large-scale neuronal theories of the brain*, pp.61-74.

Darian-Smith C., Darian-Smith I., Burman K., Ratcliffe N. (1993). Ipsilateral cortical projections to areas 3a, 3b, and 4 in the macaque monkey. *J Comp Neurol*, v.335, pp.200–213.

Dautenhahn K. and Nehaniv C.K. (2002). *Imitation in animals and artifacts*. Cambridge MA: MIT Press.

Dautenhahn, K. (1996). Embodied cognition in animals and artifacts *Proc. AAAI FS Embodied Cognition and Action*, pp.27-32.

Dayan P. and Abbott L. F. (2001). *Theoretical Neuroscience: Computational and Mathematical Modeling of Neural Systems*. MIT Press, Cambridge, MA.

de Vries J.I.P., Visser G.H.A. and Prechtl H.F.R. (1982). The emergence of fetal behaviour. I. Qualitative aspects. Early Hum Dev, v.7, pp.301–322.

Deacon T. (1997). *The symbolic species*, Norton New York.

Dean J. (1991). A model of leg coordination in the stick insect, Carausius morosus. II. Description of the kinematic model and simulation of normal step patterns. *Biological Cybernetics,* v.64, pp.403–11.

Decety J., Grezes J., Costes N., Perani D., Jeannerod M., Procyk E., Grassi F., Fazio F. (1997). Brain activity during observation of actions. Influence of action content and subject's strategy. *Brain* v.120, pp.1763–1777.

Decety J. and Ingvar D.H. (1990). Brain structures participating in mental simulation of motor behavior: A neuropsychological interpretation, *Acta Psychol*, v.73, pp.13-34.

Decety, J., Perani, D., Jeannerod, M., Bettinardi, V., Tadary, B., Woods, R., Mazziotta, J.C. and Fazio, F., (1994). Mapping motor representations with PET. Nature, v.371, pp.600-602

Decety, J., Philippon, B. and Ingvar, D.H. (1988). rCBF landscapes during motor performance and motor ideation of a graphic gessture. *Eur. Arch. Psychiatry Neurol. Sci*., v.238, pp.33-38.

Degallier S. and Ijspeert A. (2010). Modeling discrete and rhythmic movements through motor primitives: a review, Biological Cybernetics, v.103:4, pp.319-338, Springer.

Del Giudice, M., Manera, V., and Keysers, C., 2009. “Programmed to learn? The ontogeny of mirror neurons”. Developmental Science v.12. pp.350–363

Delorme A. and Thorpe S. J., (2001). Face identification using one spike per neuron: Resistance to image degradations, *Neural Networks*., v.14, pp.795– 803.

Delvolve I., Branchereau P., Dubuc R., Cabelguen J.M. (1999). Fictive rhythmic motor patterns induced by NMDA in an in vitro brain stem-spinal cord preparation from an adult urodele. *J Neurophysiol,* v.82, pp.1074–1077.

Demiris J. and Hayes G. (2002). Imitation as a dual-route process featuring predictive and learning components: a biologically plausible computational model. *In Imitation in animals and artifacts (ed. K. Dautenhahn & C. L. Nehaniv)*, pp. 327– 361, Cambridge, MA: MIT Press.

Demiris Y, Meltzoff A, (2008) The Robot in the Crib: A developmental analysis of imitation skills in infants and robots, Infant and Child Development, 17:43-53.

Denis M. (1985). Visual imagery and the use of mental practice in the development of motor skills. Canad. *J. Applied Sport Sci.*, v.10, pp.4-16.

Deutscher J., Blake A. and Reid I. (2000). Articulated body motion capture by annealed particle filtering. *In IEEE Comput. Vision Pattern Recognition (CVPR 2000)*. Piscataway, NJ: IEEE

DeWeese M., Wehr M. and Zador A. (2003). Binary spiking in auditory cortex. *J Neurosci*, v.23, pp.7940-7949.

di Pellegrino G., Fadiga L., Fogassi L., Gallese V., Rizzolatti G. (1992). Understanding motor events: a neurophysiological study. *Exp Brain Res,* v.91, pp.176-180.

Dillmann R., Kaiser M. and Ude A. (1995). Acquisition of elementary robot skills from human demonstration. *In Int. Symp. Intell. Robotic Systems (SIRS'95)*, pp. 1–38, Pisa, Italy.

Dinstein I., Thomas C., Behrmann M., Heeger D. (2008). A mirror up to nature. *Curr Biol,* v.18, pp.13–18.

Dinstein I., Hasson U., Rubin N., Heeger D. (2007). Brain areas selective for both observed and executed movements. *J Neurophysiol*, v.98, pp.1415–1427.

Dockendorf, K.P., Park, I., H.P., Principe, J.C. and DeMarse, T.B. (2009). Liquid state machines and cultured cortical networks: The separation property, *BioSystems*, v.95:2, pp.90-97.

Douglas, R. and Martin, K. (1998). Neocortex*. In G. M. Shepherd (Ed.), The synaptic organization of the brain,* v.4, pp.459–509, New York: Oxford University Press

Doya K., Katagiri K., Wolpert D. and Kawato M. (2000). Recognition and imitation of movement patterns by a multiple predictor–controller architecture. *Technical Report of IEICE TL2000-11*, pp.33–40.

Doya K. (1999). What are the computations of the cerebellum, the Basal Ganglia and the cerebral cortex

Driskell J.E., Copper C., Moran A. (1994). Does mental practice enhance motor performance. *Appl Psycho,* v.79, pp.481-492.

Duda R. O., P. E. Hart and Stork, D.G. (2000). Pattern Classification. Wiley. (2000)

Dum R. P. and Strick P. L. (1991). The origin of corticospinal projections from the premotor areas in the frontal lobe. *J. Neurosci.* v.11, pp.667–669.

Durstewitz D., Seamans J. et al. (2000). Neurocomputational models of working memory. *Nature Neuroscience,* v.3, pp.1184-1191.

Egstrom, G. H. (1964). Effects of an emphasis on conceptualizing techniques during early learning of a gross motor skill. Research Quarterly, v.5, pp.472-481.

Ekeberg Ö., Wallen P., Lanser, A., Traven H., Brodin L. and Grillner S. (1991). A computer-based model for realistic simulations of neural networks. *Biol. Cybern.*, v.65, pp.81-90.

Eliasmith C. and Anderson C. H. (2003). *Neural Engineering: Computation, Representation and Dynamics in Neurobiological Systems*. MIT Press.

Elshaw M., Weber C., Zochios A. and Wermter S. (2004). An associator network approach to robot learning by imitation through vision, motor control and language. *International joint conference on neural networks*, Budapest, Hungary.

Engelbrecht A.P. (2007). Computational intelligence: an introduction, Wiley.

Erlhagen, W., Mukovskiy A., et al. (2006). A dynamic model for action understanding and goal-directed imitation. *Brain Research*, pp.174-188.

Evangeliou M., Raos V., Galleti, C. and Savaki H., (2008), Functional imaging of the parietal cortex during action execution and observation, Cerebral Cortex, v.19:3, pp.624.

Evarts E. (1968). Role of motor cortex in voluntary movements in primates. *Handbook of Physiology: Motor Control, Part. 2,* pp.1083-1120.

Fabbri-Destro, M. and Rizzolatti, G., (2008). Mirror neurons and mirror systems in monkeys and humans. Physiology 23, 171–179.

Fadiga L., Fogassi L., Pavesi G., Rizzolatti G. (1995). Motor facilitation during action observation: a magnetic stimulation study. *J Neurophysiol*, v.73, pp.2608 –2611.

Fadiga, L., Buccino, G., Craighero, L., Fogassi, L., Gallese V. and Pavesi G. (1999). Corticospinal excitability is specifically modulated by motor imagery: a magnetic stimulation study, *Neuropsychologia,* v.37, pp.147–158.

Fadiga, L., Fogassi, L., Gallese, V., & Rizzolatti, G. (2000). Visuomotor neurons: Ambiguity of the discharge or 'motor' perception? International Journal of Psychophysiology, v.35:2–3, pp.165–177.

Feldman A.G. (1966). Functional tuning of the nervous system with control of movement or maintenance of a steady posture. III. Mechanographic analysis of execution by arm of the simplest motor tasks. Biophysics, v.11, pp.766–775.

Ferrari, P.F., Gallese, V., Rizzolatti, G., Fogassi, L., 2003. "Mirror neurons responding to the observation of ingestive and communicative mouth actions in the monkey ventral premotor cortex". European Journal of Neuroscience v.17. pp.1703–1714.

Ferrari, P.F., Visalberghi, E., Paukner, A., Fogassi, L., Ruggiero, A. and Suomi, S.J., 2006. "Neonatal imitation in rhesus macaques". PLoS Biology v.4:9. pp.1501–1508.

Fernando, C. and Sojakka, S. (2003). Pattern recognition in a bucket, *Advances in ALife*, pp.588-597.

Ferrari P.F., Visalberghi E., Paukner A., Fogassi L., Ruggiero A. and Suomi S. J. (2006). Neonatal imitation in rhesus macaques. *PLoS Biol*. v.4.

Ferrari P. F., Rozzi S. and Fogassi L. (2005). Mirror neurons responding to observation of actions made with tools in monkey ventral premotor cortex. *J. Cogn. Neurosci*, v.17, pp.212–226.

Ferster D. and Spruston N. (1995). Cracking the neuronal code, Science, v.270, pp.756-757.

Field D. J. (1994). What is the goal of sensory coding? *Neural Computation*, v.6, pp.559–601.

Fieldman, J.B., Cohen, L.G., Jezzard, P., Pons, T., Sadato, N., Turner, R., LeBihan, D. and Hallet, M. (1993). Functional neuroimaging with echo-planar imaging in humans during execution and mental rehearsal of a simple motor task. Twelfth Annual Meeting of the Society of Magnetic Resonance in Medicine, pp 1416.

Fingelkuns A.A. Fingeilcurts A.l., Krause C.M., Mottonen R. and Sams M. (2003). Cortical operational synchrony during audio-visual speech integration. *Brain and Language*, v.85, pp.297-312.

Finke R.A. (1980). Levels of equivalence in imagery and perception. *Psychol. Rev.*, v.87, pp.113-132.

Fisher, R. A. (1936). The Use of Multiple Measurements in Taxonomic Problems. *Annals of Eugenics,* v.7:2, pp.179–188.

FitzHugh R. (1961). Impulses and physiological states in theoretical models of nerve membranes. *Biophys. J.*, v.65, pp.81-90.

Flament D. and Hore J. (1988). Relations of motor cortex neural discharge to kinematics of passive and active elbow movements in the monkey. *J. Neurophysiol*, v.60, pp.1268–1284.

Flanagan J.R., Vetter P., Johansson R.S., Wolpert D.M. (2003). Prediction precedes control in motor learning. *Curr. Biol,* v.13, pp.146–150.

Flash T. and Hogan N. (1985). The coordination of arm movements: an experimentally conlurmed mathematical model. I. *Neurósci*. v.5, pp.1688—1703.

Flash T. and Hochner B. (2005). Motor primitives in vertebrates and invertebrates, *Current Opinions in Neurobiology*, v.15:6, pp.660–666.

Floreano, D. and Mattiussi, C. (2008). Bio-inspired artificial intelligence: theories, methods, and technologies, MIT Press.

Fodor, J. (1982). *The modularity of mind*, MIT Press.

Fogel, D.B. (1994). An introduction to simulated evolutionary optimization, *in IEEE Trans. Neural Networks*, v.5, pp.3-14.

Freeman W., and Bathe I M. (1993). Chaotic oscillations and the genesis of meaning in cerebral cortex. *The IPSEN symposium on "Temporal Coding in the Brain"* Pans. 11 October.

Fristen K. (1997). Imaging cognitive anatomy. *Trends in Cognitive Sciences,* v.1:1, pp.21-27.

Friston K., Frith C., et al. (1993). Functional connectivity: the principal-component analysis of large (PET) data sets, *Journal of Cerebral Blood* flow, v.3, pp.5-14.

Friston K. (1997). Transients. metastability and neural dynamics, *Neurolmage*, v.5, pp.164- 171.

Friston K., Buechel C., et al. (1997). Psychophysiological and modulatory interactions in neuroimaging. *Neuroimage,* v.6:3, pp.218-229.

Frith U. and Frith C., (2003). Development and neurophysiology of mentalizing. *Philosophical Transactions of the Royal Society B: Biological Sciences*, v.358:1431, pp. 459.

Frith C.D., Frith U. (1999). Interacting minds: A biological basis. *Science,* v.286, pp.1692-1695.

Gallese V., Fadiga L., Fogassi L., Rizzolatti G. (1996). Action recognition in the premotor cortex. *Brain,* v.119, pp.593–609.

Gallese V. and Goldman A. (1998). Mirror neurons and the simulation theory of mind reading. *Trends in Cognitive Sciences*, v.12, pp.493-501.

Gallese, V, Fogassi, L, Fadiga, L, Rizzolatti, G. 2002. Action representation and the inferior parietal lobule. Attention & Performance XIX.Common Mechanisms inPerception and Action, ed.WPrinz, B Hommel, pp. 247–66. Oxford, UK: Oxford Univ. Press

Gallese V., Fadiga L., Fogassi L., Luppino G. and Murata A., (1997). A parietal-frontal circuit for hand grasping movements in the monkey: evidence from reversible inactivation experiments. *In: Experimental Brain Research Series. Parietal Lobe Contributions to Orientation in 3D Space Edited by P. Thier and H. O. Karwath*. Berlin: Springer-Verlag, v.25, pp.255–270.

Gallese V., Keysers C. and Rizzolatti G. (2004). A unifying view of the basis of social cognition. *Trends in Cognitive Sciences*, v.8, pp.396-403.

Gandolfo F., Mussa-Ivaldi F.A., Bizzi E. (1996). Motor learning by field approximation. *Proc Natl Acad Sci,* v.93, pp.3843–3846.

Gallagher S. (2000). Philosophical concepts of the self: Implications for cognitive sciences. *Trends in Cognitive Sciences*, v.4, pp.14–21.

Gaudiano P, Grossberg S. (1992). Adaptive vector integration to endpoint: Self-organizing neural circuits for control of planned movement trajectories. Hum Mov Sci, v.11:1–2, pp.141–155.

Georgopoulos A.P., Kettner R.E. and Schwartz A.B. (1988). Primate motor cortex and free arm movements to visual targets in three-dimensional space. II. Coding of the direction of movement by a neuronal population. *Journal of Neuroscience*, v.8:8, pp.2913-2927.

Georgopoulos A.P., Caminiti R. and Kalaska J.F. (1984). Static spatial effects in motor cortex and area 5: Quantitative relations in a two-dimensional space. *Exp. Brain Res*. v.54, pp.446-454.

Georgopoulos, A.P. and Kalaska, J.F. and Caminiti, R. and Massey, J.T. (1982). On the relations between the direction of two-dimensional arm movements and cell discharge in primate motor cortex, Journal of Neuroscience, v.2:11, pp.1526-1537.

Gerstner, W. and Kistler, W.M. (2002). Spiking neuron models: Single neurons, populations, plasticity, Cambridge University Press.

Gerstner, W., Kempter, R., van Hemmen, J.L. and Wagner, H. (1996). A neuronal learning rule for sub-millisecond temporal coding, *Nature,* v.383:6595, pp.76-78.

Gerstner, W. (1998). Spiking neurons. *In Pulsed Neural Networks, edited by W. Maas and C. Bishop*, Cambridge: MIT Press.

Giszter, S.F, Mussa-Ivaldi, F.A. and Bizzi, E. (1993). Convergent force fields organized in the frog's spinal cord, *The journal of Neuroscience*, v.13:2, pp.467.

Goldman, A. (2006). Simulating minds: The philosophy, psychology, and neuroscience of mindreading. New York: Oxford University Press
Goodbody S.J. and Wolpert D.M. (1998). Temporal and amplitude generalization in motor learning. *J Neurophysiol*, v.79, pp.1825–1838.

Goodale M.A. and Milner A.D. (1992). Separate visual pathways for perception and action, Trends in Neurosciences, v.15:1, pp.20-25.

Goodman, E. and Ventura, D. (2006). Spatiotemporal pattern recognition via liquid state machines, Intl. Joint Conf. Neural Networks, pp.3848-3853.

Goldman A.I. and Sripada C.S. (2005). Simulationist models of face-based emotion recognition. *Cognition*, v.94:3, pp.193-213, Elsevier.

Gomi H. and Kawato M. (1996). Equilibrium-point control hypothesis examined by measured arm stiffness during multijoint movement, Science, v.272:5258, pp.117, American Association for the Advancement of Science.

Gopnik, A. and Wellman, H.M. (1994). The theory theory, Cambridge University Press.

Gordon R.M. (1986). Folk psychology as simulation. Mind Lang 1:158-171

Gordon J., Ghilardi M.F. and Ghez C. (1994). Accuracy of planar reaching movements: I. Independence of direction and extent variability. *Exp Brain Res*, v.99:1, pp.92-111, Berlin, Springer-Verlag.

Gordon J. Ghilardi M.F. and Ghez C. (1994). Accuracy of planar reaching movements: I. Independence of direction and extent variability. *Exp Brain Res*, v.99:1, pp.112-130.

Graybiel A. M. (1995). Building action repertories: memory and learning functions of the basal ganglia. *Current Opinion in Neurobiology*, v.5, pp.733–741.

Greenwald (1970). Sensory feedback mechanisms in performance control: with special reference to the ideo-motor mechanism. *Psychol. Rev*, v.77, pp.73–99.

Grezes J. and Decety J. (2002). Does visual perception of object afford action? Evidence from a neuroimaging study. *Neuropsychologia,* v.40, pp.212–222.

Grezes J. (1998). Top down effect of strategy on the perception of human biological motion: A PET investigation. Cognitive Neuropsychology, v.15:6, pp.553-582.

Grillner S. (2006). Biological pattern generation: the cellular and computational logic of networks in motion. *Neuron,* v.52:5, pp.751–766.

Grill-Spector K., Henson R., Martin A. (2006). Repetition and the brain: neural models of stimulus-specific effects. *Trends Cogn Sci*, v.10, pp.14–23.

Grudic G. Z. and Lawrence P. D. (1996). Human-to-robot skill transfer using the SPORE approximation. Robotics and Automation, 1996. Proceedings., 1996 IEEE International Conference on, v.4, pp.2962-2967.

Guenter F., Hersch M., Calinon S., Billard A. (2007). Reinforcement learning for imitating constrained reaching movements. *Advanced Robotics, Special Issue on Imitative Robots*, v.21:13, pp.1521-1544.

Halsband U. and Passingham R. (1985). Premotor cortex and the conditions for movement in monkeys(Macaca fascicularis). *Behavioural brain research*, v.18:3, pp.269-277.

Haralick, R.M. and Shapiro L.G. (1992). *Computer and Robot Vision*, v.1, pp. 28-48, Addison-Wesley.

Hari R., Forss N., Avikainen S., Kirveskari E., Salenius S., Rizzolatti G.(1998). Activation of human primary motor cortex during action observation: a neuromagnetic study. *Proc Natl Acad Sci,* v.95, pp.15061–15065.

Harris, C. and Stephens, M. (1988). A combined corner and edge detector, *Proceedings of the 4th Alvey Vision Conference,* pp.147-151.

Harnad, S. (1990). The symbol grounding problem, Physica D: Nonlinear Phenomena, v.42:1-3, pp.335-346, Elsevier.

Harris C.M. and Wolpert D.M. (1998). Signal-dependent noise determines motor planning. *Nature,* v.394, pp.780–784.

Haruno M., Wolpert D. M., and Kawato M. (2001). Mosaic model for sensorimotor learning and control. *Neural Computation*, v.13:10, pp.2201–2220.

Hassoun M. (1993). *Associative neural memories: Theory and implementation*. New York: Oxford University Press.

Hata T., Kanenishi K., Akiyama M., Tanaka H. and Kimura K. (2005). Real-time 3-D sonographic observation of fetal facial expression. J*. Obstet. Gynaecol*. *Res*. v.31, pp.337–340.

Hayes G. and Demiris J. (1994). A robot controller using learning by imitation. *In Proc. 2nd Int. Symp. Intell. Robotic Systems (ed. A. Borkowski & J. L. Crowley)*, pp. 198–204, Grenoble, France: LIFTA-IMAG.

Heal J. (1986). Replication and functionalism. Cambridge University, Press, Cambridge.

Hermsdorfer J., Goldenberg G., Wachsmuth C., Conrad B., Ceballos-Baumann A.O., Bartenstein P., Schwaiger M. and Boecker H. (2001). Cortical correlates of gesture processing: clues to the cerebral mechanisms underlying apraxia during the imitation of meaningless gestures, *Neuroimage*, v.14:1, pp.149-161.

Heijst J.J., Vos J.E. and Bullock D. (1998). Development in a biologically inspired spinal neural network for movement control, *Neural Networks*, v.11:7-8, pp.1305-1316.

Heuer, H.A. (1989). Multiple-representations' approach to mental practice of motor skills, *In B. Kirkcaldy (Ed). Normalities and Abnormalities in Human movement*, *v.29. Med Sport Science, Basel, Karger*, pp.36–57.

Heyes, C.M. and Ray, E.D. 2000. "What is the significance of imitation in animals?". Advances in the Study of Behavior. v.29, pp.215–245.

Heyes C. (2002). Transformational and associative theories of imitation. *In: Imitation in animals and artifacts*, pp.501-523, Cambridge, MA, US: MIT Press.

Heyes, C.M. (2001). Causes and consequences of imitation, *Trends in Cognitive Sciences,* v.5, pp.245–261.

Hodgkin A.L. and Huxley A.F. (1952). A quantitative description of membrane current and its application to conduction and excitation in a nerve. *J. of Physiol.*, v.117, pp.500-544.

Hoft B. and Arbib M. A. (1991). A model of the effects if speed accuracy and perturbation on visually guided reaching. *In R. Caminiti (Ed.), Control of arm movement in space: Neurophysiological and computational approaches; Experimental Research Series,* New York: Springer-Verlaq.

Homan M., Doucet A., de Freitas N., Jasra A. (2007). Bayesian policy learning with transdimensional MCMC. *In: Advances in Neural Information Processing Systems,* v.20, pp.665-672, (NIPS),Vancouver, BC, CA.

Hopfield J.J. (1995). Pattern recognition computation using action potential timing for stimulus representation, *Nature*, v.376, pp.33–36.

Horwitz B. (1994). Data analysis paradigms for metabolic-flow data: combining neural modeling and functional neuroimaging. *Human Brain Mapping*, v.2:1-2, pp.112-122.

Horwitz B., Soncrant T., et al. (1992). Covariance analysis of functional interactions in the brain using metabolic and blood flow data. *Nato ASI series D, Behavioral and Social Sciences*, v.68, pp.189-189.

Horwitz B., Grady C., et al. (1992). Functional associations among human posterior extrastriate brain regions during object and spatial vision. *Journal of Cognitive Neuroscience,* v.4:4, pp.311-322.

Horwitz, B. (1989). Functional neural systems analyzed by use of interregional correlations of glucose metabolism, *Visuomotor Coordination,* pp.873-892.

Houk J. C. and Wise S.P. (1995). Distributed modular architectures linking basal ganglia, cerebellum, and cerebral cortex: their role in planning and controlling action. *Cerebral Cortex*, v.5:2, pp.95.

Hourdakis E. and Trahanias P (in press, a). Improving the performance of Liquid State Machines based on the separation property, *under review by the Journal of Neurocomputing*.

Hourdakis E. and Trahanias P. (in press, b). "Observational learning inspired by human primates", Adaptive Behavior, *under review by the journal of Adaptive Behavior*.

Hourdakis E. Savaki E. and Trahanias P. (2011). “Computational modeling of cortical pathways involved in action execution and action observation”, Neurocomputing, vol. 74 , Issue 7, pp.1135-1155.

Hourdakis E. and Trahanias P. (2011a). ”Improving the performance of liquid state machines based on the separation property”, *Engineering Applications of Neural Networks, EANN11, Corfu, Greece, 2011.*

Hourdakis E. and Trahanias P. (2011b). “Observational learning based on models of overlapping pathways”, *International Conference on Artificial Neural Networks, ICANN11,* Helsinky, Finland, 2011

Hourdakis E. and Trahanias P. (2011c). “Observational learning based on overlapping pathways”, *Second International Conference on Morphological Computation,* MORPHCOMP11, Venice, Italy.

Hourdakis E. and Trahanias P. (2011d). “Computational modeling of online reaching”, *European Conference on Artificial Life,* ECAL11, Paris, France.

Hourdakis E. and Trahanias P. (2008). “A framework for automating the construction of computational models, *Congress on evolutionary computation*, *Norway* 2008.

Hourdakis, E., Maniadakis M. and Trahanias P., (2007). A biologically inspired approach for the control of the hand”, *Congress on Evolutionary Computation, Singapore 2007.*

Hubel, D. (1957). Tungsten microelectrode for recording from single units. Science, v.125:3247, pp.549-550.

Hubel, D.H. and Wiesel, T.N. (1962). Receptive fields, binocular interaction and functional architecture in the cat’s visual cortex. *Journal of Physiology*, v.160, pp.106-154.

Huffman K.J., Krubitzer L. (2001). Thalamo-cortical connections of areas 3a and M1 in marmoset monkeys. *J Comp Neurol,* v.435, pp.291–310.

Hultborn H., Illert M. and Santini M. (1976). Convergence of interneurones mediating the reciprocal Ia inhibition of motoneurons I. Disynaptic Ia inhibition of Ia inhibitory interneurones. *Acta Physiolog. Scand*, v.96, pp.193–201.

Humphrey, D. R. amd Reed, D. J. (1983). In Advances in Neurology: Motor Control Mechanisms in Health and Disease (ed. Desmedt, J. E.), pp.347–372, Raven, New York.

Hutchison W.D. (1999). Pain-related neurons in the human cingulated cortex. *Natural Neuroscience*, v.2, pp.403-405.

Iacoboni M. (2003). Understanding others: Imitation, language, empathy. *In: Hurley S and Chater N (eds). Perspectives on Imitation: From Cognitive Neuroscience to Social Science, Volume 1: Mechanisms of Imitation and Imitation in Animals*. Cambridge, MA: MIT Press.

Iacoboni M. (2005). Understanding Others: Imitation, Language, Empathy. *Perspectives on imitation From cognitive neuroscience to social science,* v.1:2, pp.77–99, Publisher: MIT Press.

Iacoboni M., Woods R.P., Brass M., Bekkering H., Mazziotta J.C. and Rizzolatti G. (1999). Cortical mechanisms of human imitation. *Science,* v.286, pp.2526–2528.

Iacoboni M., Molnar-Szakacs I., Gallese V., Buccino G., Mazziotta J. C. and Rizzolatti G. (2005). Grasping the intentions of others with one's own mirror neuron system. *PLoS Biology*, v.3:3, p.79.

Iacoboni M. (2009). Imitation, empathy, and mirror neurons. *Annu. Rev. Psychol*. v.60, pp.653–670.

Iacoboni M. (2005). Neural mechanisms of imitation. *Curr. Opin. Neurobiol*. v.15, pp.632–637.

Ijspeert J.A., Nakanishi J. and Schaal S. (2002a). Learning rhythmic movements by demonstration using nonlinear oscillators. *In IEEE Int. Conf. Intell. Robots Systems (IROS 2002)*, Lausanne, Piscataway, NJ: IEEE.

Ijspeert J. A., Nakanishi J. and Schaal S. (2002b). Movement imitation with nonlinear dynamical systems in humanoid robots. *In Int. Conf. Robotics Automation (ICRA2002)*, pp.1398-1403, Washington, DC, Piscataway, NJ: IEEE.

Ijspeert A.J., Nakanishi J., and Schaal S. (2003). Learning attractor landscapes for learning motor primitives, *in Advances in Neural Information Processing Systems (NIPS), S. Becker, S. Thrun, and K. Obermayer, Eds.,* v.15, pp. 1547–1554, Cambridge, MA: MIT Press.

Ijspeert A.J., Crespi A., Ryczko D. and Cabelguen J.M. (2007). Rom swimming to walking with a salamander robot driven by a spinal cord model, *Science,* v.315:5817, pp.1416.

Inamura, T., et al. (2001). Imitation and primitive symbol acquisition of humanoids by the integrated mimesis loop. *Robotics and Automation, 2001. Proceedings 2001 ICRA. IEEE International Conference on*, v.4, pp.4208-4213.

InamuraT, Nakamura Y, Toshima I. (2001). Embodied Symbol Emergence based on Mimesis Theory, International Journal of Robotics Research, 23(4), pp.363-377.

Ingvar, D.H. and Philipson, L. (1977). Distribution of cerebral blood flow in the dominant hemisphere during motor ideation and motor performance. Ann. Neurol., v.2, pp.230-237.

Ito M. (1970). Neurophysiological aspects of the cerebellar motor control system. *Int. J. Neurol*., v.7, pp.162–176.

Ito and Tani (2004). On-line imitative interaction with a humanoid robot using a dynamic neural network model of a mirror system.

Ivry R.B. (1996). The representation of temporal information in perception and motor control. *Current Opinion in Neurobiology*, v.6, pp.851–857.

Jack J.J.B., Noble D. and Tsien R.W. (1975). *Electric current flow in excitable cells*. Oxford University Press.

James W. (1890). *The principles of psychology*. Cambridge: Harvard University Press.

Jeannerod M., Arbib M.A., Rizzolatti G. and Sakata H. (1995). Grasping objects: The cortical mechanisms of visuomotor transformation, *Trends Neurosci.*, v.18:7, pp.314–320.

Jeannerod M. (1994). The representing brain: neural correlates of motor intention and imagery. Behav. *Brain Sci*. v.17, pp.187–245.

Jeannerod, M. and Johnson-Frey, SH. (2003). Simulation of action as a unifying concept for motor cognition, Taking action: Cognitive neuroscience perspectives on intentional acts, pp.139-163, MIT Press, Cambridge.

Jeannerod, M., (2003). The mechanism of self-recognition in human. Behav. Brain Res. 142, 1–15.

Jeannerod M. (1988). *The Neural and Behavioural Organization of Goal-Directed Movements*, Clarendon Press, Oxford (1988).

Jeannerod M. and Decety J (1995). Mental motor imagery: a window into the representational stages of action. Curr Opin Neurobiol, v.5, pp.727-732.

Jellema T., Baker C.I., Oram M.W. and Perrett D.I. (2002). Cell populations in the banks of the superior temporal sulcus of the macaque and imitation. *In: The imitative mind: Development, evolution, and brain bases*, pp.267-290, New York, NY, US: Cambridge University Press.

Jima V.K. and Keis A.S. (2000). Spatiotemporal pattern formation in neural systems with heterogeneous connection topologies. *Physical Review. E, Statistical, Nonlinear, and Soft Matter Physics*, v.62, pp.8462-8465.

Joel, D., Niv, Y. and Ruppin, E. (2002). Actor-critic models of the basal ganglia: a new anatomical and computational perspectives, *Neural networks*, v.16:4, pp.535-547.

John E.R. (2002). The neurophysics of consciousness. Brain Reretirch. *Brain Research Reviews*, v.39, pp.1-28.

Jones S.S. (2009). The development of imitation in infancy, *Philosophical Transactions of the Royal Society B: Biological Sciences*, v. 364:1528, pp.2325-2335, The Royal Society.

Jones E.G. and Powell T.P.S. (1969). Connections of the somatic sensory cortex of the rhesus monkey. I. Ipsilateral cortical connections, *Brain*, v.92:1969, pp.477-502.

Jordan M.I. and Rumelhart D.E. (1992). Forward models: Supervised learning with a distal teacher. *Cognitive Science*, v.16, pp.307–354.

Joshi and Maass (2005). Movement generation with circuits of spiking neurons

Kaas J.H., Nelson R., Sur M., Lin C. and Merzenich, M.M. (1979). Multiple representations of the body within the primary somatosensory cortex of primates, *Science*, v.204, pp.521-523.

Kaas J.H., Rockland K.L.S. and Peters A. (1997). *Extrastriate cortex in primates*, v.12, pp.205-241, New York: Plenum Press.

Kalaska J., Caminiti R. and Georgopoulos A. (1983). Cortical mechanisms related to the direction of two-dimensional arm movements: relations in parietal area 5 and comparison with motor cortex. *Experimental Brain Research*, v.51:2, pp.247-260.

Kandel, E.R. and Schwartz, J.H. and Jessell, T.M. et al. (2000). *Principles of neural science*, v.4, McGraw-Hill, New York.

Karni A., Meyer G., Jezzard P., Adams M.M., Turner R., Ungerleider L.G. (1995). Functional MRI evidence for adult motor cortex plasticity during motor skill learning. *Nature,* v.377, pp.155–158.

Kawamura S. and Fukao N. (1994). Interpolation for input torque patterns obtained through learning control. *In Int. Conf. Automation, Robotics Computer Vis. (ICARCV '94)*, pp. 183–191, Singapore.

Kawato M. (1999). Internal models for motor control and trajectory planning. *Curr. Opin. Neurobiol*. v.9, pp.718–727.

Kawato M. (1990). Feedback-error-learning neural network for supervised learning. *In R. Eckmiller (Ed.), Advanced neural computers,* pp.365–372, Amsterdam: North-Holland.

Kawato M., Furawaka K. and Suzuki R. (1987). A hierarchical neural network model for the control and learning of voluntary movement, Biological cybernetics, v.57:3, pp. 169-185.

Keysers, C. and Gazzola, V. (2009). Expanding the mirror: vicarious activity for actions, emotions, and sensations. *Curr. Opin. Neurobiol,* v.19, pp.666.

Kempter R., Gerstner W., van Hemmen J.L. and Wagner H. (1998). Extracting oscillations: Neuronal coincidence detection with noisy periodic spike input, *Neural Computat*., v.10, pp.1987–2017.

Kiehn O., Hounsgaard J. and Sillar K. (1997). Basic building blocks of vertebrate spinal central pattern generators, *In Neurons, Networks, and Motor Behavior (P. Stein, S. Grillner, A. Selverston, and D. Stuart, Eds.)*, pp. 47–60, Cambridge, MA: MIT Press.

Kilintari, M., Raos, V., and Savaki, E. (2010). Grasping in the dark activates early visual cortices, *Cerebral Cortex*, v.21:4, pp.949.

Kilner J., Neal A., Weiskopf N., Friston K. and Frith C. (2009). Evidence of mirror neurons in human inferior frontal gyrus. J Neurosci, v.29, pp.10153–10159.

Kleim J.A., Barbay S. and Nudo R.J. (1998). Functional reorganization of the rat motor cortex following motor skill learning. *J Neurophysiol,* v.80, pp.3321–3325.

Knudsen E.I. and Konishi M. Mechanisms of sound localization in the barn owl (1979). J. Comp. Physiol. v.133, pp.13–21.

Koch C. (1999). Biophysics of computation. Oxford University Press.

Kohler, E., Keysers, C., Umilta, M. A., Fogassi, L., Gallese, V. and Rizzolatti, G. (2002). Hearing sounds, understanding actions: Action representation in mirror neurons. *Science*, v.297:5582, pp.846–848.

Kohonen T. (1984). *Sell-organization and associative memory*. Berlin: Springer.

Kok, S, (2007). *Liquid State Machine Optimization*, Master Thesis, Utrecht University

Konig, P., Engel, A. K., and Singer, W. (1996), Integrator or coincidence detector? the role of the cortical neuron revisited, TINS, v.19. pp.130–137.

Kosslyn, S.M., Flynn, R.A., Amsterdam, J.B. and Wang, G. (1990). Components of highlevel vision: A cognitive neuroscience analysis and accounts of neurological syndromes. *Cognition*, v.34, pp.203-277.

Krakauer J.W., Ghez C., Ghilardi M.F. (2005). Adaptation to visuomotor transformations: consolidation, interference, and forgetting. *J Neurosci*, v.25, pp.473– 478.

Krubitzer L., Huffman K.J., Disbrow E. and Recanzone G. (2004). Organization of area 3a in Macaque monkeys: Contributions to the cortical phenotype, *Journal of comparative neurology,* v.471:1, pp.97-111.

Kuniyoshi Y., Yorozu Y., Inaba M. and Inoue H. (2003). From visuo-motor self-learning to early imitation—a neural architecture for humanoid learning. *International conference on robotics & automation*, Taipei, Taiwan.

Kuperstein M. (1988). Neural model of adaptive hand-eye coordination for single postures. *Science*, v.239, pp.1308–1311.

Kurata K. (1991). Corticocortical inputs to the dorsal and ventral aspects of the premotor cortex of macaque monkeys. Neurosci. Res. v.12, pp.263–280.

Kurata K. and Tanju, J. (1986). Premotor cortex neurons in macaques: activity before distal and proximal forelimb movements. *J. Neurosci,* v.6, pp.403–411.

Kwan H.C., Mackay W.A., Murphy J.T. and Wong Y.C. (1986). Properties of visual cue responses in primate precentral cortex. *Brain Res*. v.343, pp.24– 35.

Lackner J.R. and Dizio P. (1998). Adaptation in a rotating artificial gravity environment, *Brain Res. Rev*. v.28, pp.194–202.

Lackner J.R., Dizio P. (1994). Rapid adaptation to Coriolis force perturbations of arm trajectory. *J Neurophysiol,* v.72, pp.299 –313.

Lacquaniti F. et al. (1995). Representing spatial information for limb movement: role of area 5 in the monkey. *Cerebral Cortex*, v.5:5, pp.391-409.

Legenstein, R. and Maass, W. (2007). Edge of chaos and prediction of computational performance for neural circuit models, *Neural Nets*, v.20:3, pp.323-334.

Lepage, J.F., and Théoret, H. (2007). "The mirror neuron system: grasping others' actions from birth?" Developmental Science, v.10, pp.513–524.

Leslie A. and Thaiss L. (1992). Domain specificity in conceptual development: Neuropsychological evidence from autism. *Cognition*, v.43:3, pp.225-251.

Levins R. (1966). The strategy of model building in population biology. *American Scientist,* v.54, pp.421–31.

Liberman, Alvin M. and Whalen D.H. (2000). On the relation of speech to language. *Trends in Cognitive Sciences*, v.4, pp.187-196.

Lin L.J. (1991). Programming robots using reinforcement learning and teaching. *In Proc. 9th Natl Conf. Artificial Intell.*, v.2, pp. 781–786, Anaheim, CA, Menlo Park, CA: AAAI.

Linsker, R. (1986). From basic network principles to neural architecture. *Proceedings of the National Academy of Sciences,* v.83, pp.7508–7512.

Lozano-Perez T. (1982). Task planning. *In: Brady M, Hollerbach JM, Johnson TL, Lozano-Perez T, Mason MT (eds) Robot motion: planning and control*. MIT Press, Cambridge.

Maass W. (1997). Fast sigmoidal networks via spiking neurons, *Neural Comp*., v.9:2, pp. 279–304.

Maass W. (1997). *The Third Generation of Neural Network Models*, Technische Universität Graz.

Maass W., Schnitger G. and Sontag E. (1991). On the computational power of sigmoid versus boolean threshold circuits, *Proc. of the 32nd Annual IEEE Symposium on Foundations of Computer Science*, pp. 767-776.

Maass, W. and Natschlager, T. and Markram, H. (2003). A model for real-time computation in generic neural microcircuits, pp.229-236.

Maass W. (1997b). Fast sigmoidal networks via spiking neurons. *Neural Computation*, v.9, pp.279–304.

MacKay D.M. and McCulloch W.S. (1952). The limiting information capacity of a neuronal link, Bull. *Math. Biophys*., v.14, pp.127–135.

Maeda F., Kleiner-Fisman G. and Pascual-Leone A. (2002). Motor facilitation while observing hand actions: specificity of the effect and role of observer's orientation. *J Neurophysiol,* v.87, pp.1329 –1335. *Exp Brain Res,* v.112, pp.103–111.

Mainen Z.F. and Sejnowski T. J. (1995). Reliability of spike timing in neocortical neurons. Science, v.268, pp.1503–1506.

Maniadakis M., Hourdakis E. and Trahanias P. (2007). Modeling overlapping action execution/observation brain pathways, International Joint Conference on Artificial Neural Networks, IJCAI, Atlanta 2007.

Manwani and Koch (1999). Detecting and Estimating Signals in Noisy Cable Structures, I: Neuronal Noise Sources, Neural Computation, v.11:8, pp.1797-1829.

Markram H., Wang, Y. and Tsodyks, M. (1998). Differential signalizing via the same axon of neocortical pyramidal neurons, *Proceedings of the National Academy of Sciences*, v.95:9, pp.5323.

Mataric M. and Pomplun M. (1998). Fixation behavior in observation and imitation of human movement. *Cogn. Brain Res*, v.7, pp.191–202.

Mataric M. and Cliff D. (1997). Challenges in evolving controllers for physical robots. Robotics and autonomous systems, v.19:1, pp.67-84.

Mataric M. (2001). Visuo-motor primitives as a basis for learning by imitation. *In Kerstin Dautenhahn and Chrystopher Nehaniv, editors, Imitation in Animals and Artifacts.* MIT Press.

Matelli M., Luppino G., Murata A. and Sakata H. (1994). Independent anatomical circuits for reaching and grasping linking the inferior parietal sulcus and inferior area 6 in macaque monkey. *Soc. Neurosci. Abstr*. v.20, pp.404.4.

Matser, J. (2010). *Structured liquids in liquid state machines*, Master Thesis, Utrecht University

Mattar A.G. and Gribble P.L. (2005). Motor learning by observation. *Neuron*, v.46, pp.153–160.

Mattson, J. and M. Simon (1996). *The pioneers of NMR and magnetic resonance in medicine: the story of MRI*, Bar-Ilan University Press.

Maynard Smith J. (1974). *Models in ecology*. Cambridge University Press.

McIntosh A.R. and Gonzalez-Lima F. (1994). Structural equation modeling and its application to network analysis in functional brain imaging, *Human Brain Mapping,* v.2:1-2, pp.2-22.

McIntosh, A.R. and Gonzalez-Lima, F. (1991). Structural modeling of functional visual pathways mapped with 2-deoxyglucose: effects of patterned light and footshock, v.578:1-2, pp.75-86.

McIntosh, A., C. Grady, et al. (1994). Network analysis of cortical visual pathways mapped with PET. *Journal of Neuroscience*, v.14:2, pp.655-666.

McKiernan B.J., Marcario J.K., Karrer J.H. and Cheney P.D. (1998). Corticomotoneuronal postspike effects in shoulder, elbow, wrist, digit, and intrinsic hand muscles during a reach and prehension task. *J. Neurophysiol*, v.80, pp.1961–1980.

Mehta B., Schaal S. (2002). Forward models in visuomotor control. *J. Neurophysiol*, v.88, pp.942–953.

Mehta, M.R. and Wilson, M.A. (1999). From hippocampus to V1: Effect of LTP on spatio-temporal dynamics of receptive fields, *Computational Neuroscience,* v.32, pp.905-911.

Meltzoff A.N. and Moore M.K. (1994). Imitation, memory, and the representation of persons. *Infant Behav. Dev*. v.17, pp.83–99.

Meltzoff, A.N. and Decety, J., 2003. “What imitation tell us about social cognition: a rapprochement between developmental psychology and cognitive neuroscience”. Philosophical Transactions of the Royal Society London B, Biological Sciences v.358, pp.491–500.

Meltzoff A. and Moore M. (1995). Infants understanding of people and things: From body imitation to folk psychology. *The body and the self*, pp.43-69.

Meltzoff, A.N. (2002) Imitation as a mechanism of social cognition: Origins of empathy, theory of mind, and the representation of action. *In Handbook of Childhood Cognitive Development (Goswami, U., ed.)*, Blackwell Publishers

Meltzoff, A.N. and Gopnik, A. (1993). The role of imitation in understanding persons and developing a theory of mind, *in Understanding Other Minds (Baron-Cohen, S., Tager-Flusberg, H. and Cohen, D.J., eds)*, pp. 335–366, Oxford Medical Publications.

Meltzoff A.N. and Moore M.K. (1977). Imitation of facial and manual gestures by human neonates. *Science,* v.198:4312, pp.75–78.

Mendell L. M. (1984). Modifiability of spinal synapses. *Physiolog. Rev.*, v.64, pp.260–324.

Merzenich M.M., Nelson R.J., Stryker M.P., Cynader M.S., Schoppmann A., Zook J.M. (I984). Somatosensory cortical map changes following digit amputation in adult monkeys. *J. Comp. Neurol*, v.224:591.

Mesulam M. (1990). Large-scale neurocognitive networks and distributed processing for attention, language, and memory. *Annals of Neurology,* v.28:5, pp.597-613.

Metta, G., Sandini G. and Konczak J. (1999). A developmental approach to visually-guided reaching in artificial systems. *Neural networks*, v.12:10, pp 1413-1427.

Miall C. (1989). The diversity of neuronal properties. *In: The computing neuron, ed. R. Durbin, C. Miall & G. Mitchison*. Addison-Wesley.

Miall R.C. and Wolpert D.M. (1996). Forward models for physiological motor control. *Neural Networks,* v.9, pp.1265–1285.

Miall R.C., Weir D.J., Wolpert D.M. and Stein J.F. (1993). Is the cerebellum a Smith predictor?. *J. Motor Behav.*, v.25:3, pp.203–216.

Miall RC and Wolpert DM. (1996). Forward models for physiological motor control. *Neural Networks,* v.9, pp.1265–1279.

Michel, P. and Gold, K. and Scassellati, B.(2004) Motion-based robotic self-recognition, Intelligent Robots and Systems, 2004.(IROS 2004). Proceedings. 2004 IEEE/RSJ International Conference on, v.3, pp.2763-2768

Mikami A. and Kubota K. (1980). Inferotemporal neuron activities and color discrimination with delay. *Brain Res,* v.182, pp.65–78.

Miller W.T. (1987). Sensor-based control of robotic manipulators using a general learning algorithm. *IEEE J. Robotics and Automation*, v.3, pp.157– 165.

Mitz A., Godschalk M. and Wise S. (1991). Learning-dependent neuronal activity in the premotor cortex: activity during the acquisition of conditional motor associations. *Journal of Neuroscience*, v.11:6, pp.1855-1872.

Miyachi S., Hikosaka O., Miyashita K., Karadi Z. and Rand M.K. (1997). Differential roles of monkey striatum in learning of sequential hand movement. *Experimental Brain Research*, v.151, pp.1–5.

Miyamoto H. and Kawato M. (1998). A tennis serve and upswing learning robot based on bi-directional theory. *Neural Networks,* v.11, pp.1331–1344.

Miyamoto, H., Schaal, S., Gandolfo, F., Koike, Y., Osu, R., Nakano, E., Wada, Y. and Kawato, M. (1996). A Kendama learning robot based on bi-directional theory. Neural Networks, v.9, pp.1281–1302.

Myowa-Yamakoshi, M. and Matsuzawa, T. (1999). Factors Influencing Imitation of Manipulatory Actions in Chimpanzees (Pan troglodytes), *Journal of Comparative Psychology*, v.113:2, pp.128-136, Elsevier.

Montague P., Berns G., Cohen J., McClure S., Pagnoni G., Dhamala M., Wiest M., Karpov I., King R., Apple N. and Fisher R. (2002). Hyperscanning: Simultaneous fMRI during Linked Social Interactions. *Neuroimage,* v.16:1159.

Montague P.R., Dayan P. and Sejnowski T.J. (1996). A framework for mesencephalic dopamine systems based on predictive Hebbian learning. Journal of Neuroscience, v.16, pp.1936–1947.

Mountcastle V. et al. (1975). Posterior parietal association cortex of the monkey: command functions for operations within extrapersonal space. *Journal of Neurophysiology*, v.38:4, pp.871-908.

Mountcastle V.B. (1978). An organizing principle for cerebral function: The unit module and the distributed system. *In The Mindful Brain, ed. G. M. Edelman, V. B. Mountcastle,* Cambridge: MIT Press.

Murata A, et al. (1997). Object representation in the ventral premotor cortex (area F5) of the monkey. *Am Physiological Soc*, pp.2226-2230.

Murata, V., Gallese, G., Luppino et al. (2000). Selectivity for the Shape, Size, and Orientation of Objects for Grasping in Neurons of Monkey Parietal Area AIP, *Journal of Neurophysiology*, v.83:5, pp.2580-2601.

Murphy J.T., Wong Y.C. and Kwan H. C. (1985). Sequential activation of neurons in primate motor cortex during unrestrained forelimb movement. *J. Neurophysiol,* v.53, pp.435-445.

Mussa-Ivaldi F. A. and Bizzi E. (2000). Motor learning through the combination of primitives, *Phil. Trans. R. Soc. Biol. Sci*. v.355, pp.1755–1769.

Mussa-Ivaldi E A., Giszter S. F and Bizzi E. (1994). Linear combination of primitives in vertebrate motor control. Proc. Nut. Acad. Sci,. v.91, pp.7534-7538.

Myowa M. (1996). Imitation of facial gestures by an infant chimapnzee. Primates, v.37, pp.207–213.

Nagell K., Olguin R.S. and Tomasello M. (1993). Processes of social learning in the tool use of chimapnzees (Pan troglodytes) and human children (Homo sapiens). *J. Comp. Psychol,* v.107, pp.174–186.

Nagumo J.S., Arimoto S. and Yoshizawa S. (1962). An active pulse transmission line simulating nerve axon. *Proc. IRE*,v. 50, pp2061-2070.

Nehaniv and Dautenhahn K. (1999). Of hummingbirds and helicopters: an algebraic framework for interdisciplinary studies of imitation and its applications. *In Learning robots: an interdisciplinary approach (ed. J. Demiris & A. Birk)*, Singapore: World Scientific.

Nehaniv C.L. and Dautenhahn K. (2002). The correspondence problem, *Imitation in animals and artifacts*, The MIT Press.

Nelson W.L. (1983). Physical principles for economies of skilled movements, *Biological Cybernetics*, v.46:2, pp.135-147.

Newell A. and Simon H. (1976). Computer science as empirical inquiry: Symbols and search. Communications of the ACM, v.19:3, pp.113-126.

Newlin D.B. (2009). *The Human Mirror Neuron System, Handbook of research on agent-based societies: social and cultural interactions*, Information Science Publishing.

Nichols T.R. (1994). A biomechanical perspective on spinal mechanisms of coordinated musular action: an architecture principle. *Acta Anatomica*,v. 151,pp. 1–13.

Nishitani N., Hari R. (2000). Temporal dynamics of cortical representation for action. Proc Natl Acad Sci, v.97, pp.913–918.

Nishitani N., Hari R. (2002). Viewing lip forms: cortical dynamics. Neuron, v.36, pp.1211–1220.

Nolfi S. and Floreano D. (2004). *Evolutionary robotics: The biology, intelligence, and technology of self-organizing machines,* Bradford Book.

Nordlie E., Gewaltig M-O, Plesser H.E. (2009). Towards Reproducible Descriptions of Neuronal Network Models. PLoS Comput Biol, v.5.

Northoff G and Bermpohl F, (2004) Cortical midline structures and the self, TRENDS in Cognitive Sciences Vol.8 No.3.

Norton D, and Ventura D, (2010). Improving the performance of liquid state machines through iterative refinement of the reservoir, *Neurocomputing,* v.73, pp.2893-2904.

Nowak. L. G.. and Bullier, J. (1997). The timing of information transfer in the visual system. In J. Kaas. K. Rocklund & A. Peters. Extrastriate cortex in primates (pp. 205—241). New York: Plenum Press.

Ogawa, S. and Lee, T. M. (1990). Magnetic resonance imaging of blood vessels at high fields: in vivo and in vitro measurements and image simulation. Magn. Reson. Med. 16, 9:18.

Olivier, M., (2004). Professional Mobile Robot Simulation, *Int. Journal of Advanced Robotic Systems,* v.1 pp. 39-42.

Otsu, N., (1979). A Threshold Selection Method from Gray-Level Histograms. *IEEE Transactions on Systems, Man, and Cybernetics*, v.9:1 , pp. 62-66.

Ozonoff S., Pennington B.F. and Rogers S.J. (1991). Executive function deficits in high-functioning autistic children: relationship to theory of mind, *J. Child Psychol. Psychiatry,* v.32, pp.1081–1105.

Oztop E., Wolpert D. and Kawato M. (2005). Mental state inference using visual control parameters. *Brain Research Cognitive Brain Research*, v.22:2, pp.129–151.

Oztop E., Chaminade T., Cheng G. and Kawato M. (2005). Imitation bootstrapping: Experiments on a robotic hand. *IEEE-RAS international conference on humanoid robots,* Tsukuba, Japan.

Oztop E. and Arbib M.A. (2002). Schema design and implementation of the grasp-related mirror neuron system. *Biol. Cybern*.v.87, pp.116–140.

Oztop E. and Kawato M. and Arbib M. (2006). Mirror neurons and imitation: A computationally guided review, v.19:3, pp.254-271, Elsevier.

Paine R. and Tani J. (2004). Motor primitive and sequence self-organization in a hierarchical recurrent neural network. *Neural networks*, v.17:8-9, pp.1291-1309.

Palmer, S. and R. Kimchi (1986). "The information processing approach to cognition." Approaches to cognition: Contrasts and controversies: 37-77.

Parsons L.M., Gabrieli J.D.E., Phelps E.A., Gazzaniga M.S. (1998). Cerebrally lateralized mental representations of hand shape and movement. *J Neurosci*, v.18, pp.6539-6548.

Pascual-Leone A., Nguyet D., Cohen L.G., Brasil-Neto J.P., Cammarota A., Hallett M. (1995). Modulation of muscle responses evoked by transcranial magnetic stimulation during the acquisition of new fine motor skills. *J Neurophysiol,* v.74, pp.1037–1045.

Passingham, R. (2009). How good is the macaque monkey model of the human brain?, v.19:1, pp.6-11, Elsevier.

Paulin M.G. (1993). The role of the cerebellum in motor control and perception. *Brain, Behaviour and Evolution*, v.41, pp.39–50.

Pearce, J.M., 1987. “A model of stimulus generalization in Pavlovian conditioning”. Psychological Review v.94, pp.61–73.

Penfield W. and Rasmussen T. (1938). The Cerebral Cortex of Man. *Arch Neurol Psychiatry,* v.40, pp.417-442.

Penfield W. and Rasmussen T. (1950). *The Cerebral Cortex of Man*, New York: MacMillan.

Perrett D. et al. (1989). Frameworks of analysis for the neural representation of animate objects and actions, pp.87-113.

Perrett, D.,M., Harries, et al. (1990). *Retrieval of Structure from Rigid and Biological Motion: An Analysis of the Visual Responses of Neurones in the Macaque Temporal Cortex*, John Wiley & Sons.

Perner J. (1991). *Understanding the Representational Mind*, MIT Press.

Perner J. and Lang B. (1999). Development of theory of mind and executive control, *Trends in cognitive sciences*, v.3:9, pp. 337-344.

Peters J. and Schaal S. (2007). Reinforcement learning by reward-weighted regression for perational space control. In Proceedings of the International Conference on Machine Learning (ICML), Orvallis, OR, USA.

Petrides M. and Pandya, D.N. (1997). Comparative architectonic analysis of the human and the macaque frontal cortex. *In F. Boller & J. Grafman (Eds.), Handbook of Neuropsychology*, v.9, New York: Elsevier.

Petrides M. (1982). Motor conditional associative-learning after selective prefrontal lesions in the monkey. *Behavioural brain research*, v.5:4, pp.407.

Phillips, C.G., S. Zeki, et al. (1984). Localization of function in the cerebral cortex; Past, present and future. *Brain*, v.107, pp.327-361.

Phillips, C.G., S. Zeki, and Barlow H.B. (1984). Localization of function in the cerebral cortex; Past, present and future. *Brain,* v.107: pp.327-361.

Piaget J. (1951). *Play, dreams, and imitation in childhood*. New York: Norton

Piaget J. (1962). *Dreams and imitation in childhood*, New York: W.W. Norton.

Plaut D.C. and Hinton, G.E. (1987). Learning sets of filters using backpropagation. *Computer Speech and Language,* v.2, pp.35-61.

Poggio T. and Bizzi E. (2004). Generalization in vision and motor control. *Nature,* v.431, pp.768 –774.

Poggio T. (1990). A theory of how the brain might work. *Cold Spring Harb Symp Quant Biol,* v.55, pp.899 –910.

Poggio T. and Girosi F. (1990a). Networks for approximation and learning. *Proc. IEEE,* v.78, pp.1481—1497.

Poggio T., Girosi F. (1 990b). A theory of networks for learning. *Science,* v.247, pp.978—982.

Pook P.K. and Ballard D.H. (1993). Recognizing teleoperated manipulations. *In Proc. IEEE Int. Conf. Robotics Automation*, v.3, pp.913–918, Atlanta, GA, Piscataway, NJ: IEEE

Posner M. and Raichle M. (1994). Images of mind. New York.

Posner, M., S. Petersen, et al. (1988). "Localization of cognitive operations in the human brain." Science 240(4859): 1627-1631.

Porter R. and McLewis M. (1975). Relationship of neuronal discharges in the precentral gyms of monkeys to the performance of arm movements. *Brain Res*, v.98, pp.21-36.

Pouget A., Snyder L.H. (2000). Computational approaches to sensorimotor transformations. Nat Neurosci [Suppl], v.3, pp.1192–1198.

Premack D. and Woodruff G. (1978). Does the chimpanzee have a theory of mind. *Behavioral and Brain sciences*, v.1, pp.515-526.

Prinz W. (2006). What re-enactment earns us. *Cortex,* v.42:4, pp.515–517.

Prinz W. (2002). Experimental approaches to imitation. In *Meltzoff & Prinz 2002*, pp. 143– 62.

Purves, D. and Augustine, G.J. and Fitzpatrick, D. and Katz, L.C. and LaMantia, A. and McNamara, J.O. and Williams (2001). *Neuroscience*, MA: Sinauer Associates, Sunderland.

Ramachandran, V.S., 2000. “Mirror neurons and imitation learning as the driving force behind ‘‘the great leap forward’’ in human evolution”, online: http://www.edge.org/3rd_culture/ramachandran/ramachandran_index.html

Raos, V. and Evangeliou, M.N. and Savaki, H.E. (2004). Observation of action: grasping with the mind's hand, Neuroimage, v.23:1, pp.193-201.

Raos, V. and Evangeliou, M.N. and Savaki, H.E. (2007). Mental simulation of action in the service of action perception, v.27:46, pp.12675--12683.

Richardson A. (1967). Mental practice: A review and discussion, Part 1. *Research Quarterly*, v.38, pp.95-107.

Riehle A. (1991). Visually induced signal-locked neuronal activity changes in precentral motor areas of the monkey: hierarchical progression of signal processing. *Brain Res*, v.540, pp.131–137.

Rieke, F., (1999). Spikes, Exploring the neural code, MIT Press.

Rizzolatti G., 2005. The mirror neuron system and its function in humans. *Anat. Embryol,* v.210, pp.419–421.

Rizzolatti, G. and Craighero, L.(2004). The mirror-neuron system, An. Rev. Neurosci., v.27, pp.169-192.

Rizzolatti G., Fadiga L., Fogassi L., Gallese V. (1996). Premotor cortex and the recognition of motor actions. Cogn. *Brain Res,* v.3, pp.131–41.

Rizzolatti G., Arbib M. (1998). Language within our grasp. *Trends Neurosci,* v.21, pp.188-194.

Rizzolatti G. and Fadiga L. (1998). Grasping objects and grasping action meanings: the dual role of monkey rostroventral premotor cortex (area F5). Wiley.

Rizzolatti G., Fogassi L. and Gallese V. (2001). Neurophysiological mechanisms underlying the understanding and imitation of action. *Nat. Rev., Neurosci,* v.2, pp.661–670.

Rizzolatti G. et al. (1988). Functional organization of inferior area 6 in the macaque monkey. *Experimental Brain Research*, v.71:3, pp.491-507.

Rizzolatti G. 2004. The mirror-neuron system and imitation. In Perspectives on Imitation: From Mirror Neurons to Memes, ed. S Hurley, N Chater. Cambridge, MA: MIT Press.

Roland P. (1993). *Brain activation*, Wiley-Liss New York.

Roland, P.E., Skinhoj, E., Lassen, N.A. and Larsen, B. (1980). Different cortical areas in man in organization of voluntary movements in extrapersonal space. *J. Neurophysiol*., v.43, pp.137-150.

Rohrbaugh, J., R. Parasuramanan, et al. (1993). Event-related brain potentials., *Cognitive and Behavioral Neurology*, v.6:2, pp.136.

Rolls E.T., Tovee M.J. (1995). Sparseness of the neuronal representation of stimuli in the primate temporal visual cortex. *J Neurophysiol,* v.73, pp.713-726.

Rolls E.T. (1992). Neurophysiological mechanisms underlying face processing within and beyond the temporal cortical areas, *Philos. Trans. R. Soc. Lond. B*, v.335, pp.11–21.

Rolls, E. and A. Treves (1998). *Neural networks and brain function*, Oxford Oxford, UK.

Roth, M. et al. (1996) Possible involvement of primary motor cortex in mentally simulated movement: a functional magnetic resonance imaging study, *NeuroReport,* v.7:7, pp.1280–1284.

Roy, C. and C. Sherrington (1890). On the regulation of the blood-supply of the brain. *The Journal of Physiology,* v.11:1-2, pp.85.

Rozzi S., Ferrari P.F., Bonini L., Rizzolatti G. and Fogassi L. (2008). Functional organization of inferior parietal lobule convexity in the macaque monkey: electrophysiological characterization of motor, sensory and mirror responses and their correlation with cytoarchitectonic areas. *Eur. J. Neurosci*, v.28, pp.1569–1588.

Rumelhart, David E., Geoffrey E. Hinton, and R. J. Williams. (1986). Learning Internal Representations by Error Propagation, *David E. Rumelhart, James L. McClelland, and the PDP research group. (editors), Parallel distributed processing: Explorations in the microstructure of cognition*, v.1, MIT Press.

Rumelhart, D. and J. McClelland (1987). Parallel distributed processing, exploitation in the microstructure of cognition-Vol. 1: Foundations.

Russell J. (1996). *Agency. Its Role in Mental Development*, Erlbaum.

Sackett, R. S. (1934). The influence of symbolic rehearsal upon the retention of a maze habit. *Journal of General Psychology*, v.10, pp.376-395.

Saltzman E. (1979). Levels of sensorimotor representation. *J. Math. Psychol*. v.20, pp.91–163.

Sakata, H., et al., (1973). Somatosensory properties of neurons in the superior parietal cortex (area 5) of the rhesus monkey. *Brain Research*, v.64, pp.85.

Sakata, H., et al. (1995). Neural mechanisms of visual guidance of hand action in the parietal cortex of the monkey. *Cerebral Cortex*, v.5:5, pp.429-438.

Saltzman E. and Kelso S.J.A. (1987). Skilled actions: a task dynamic approach. *Psychol. Rev*, v.94, pp.84–106.

Sammut C., Hurst S., Kedzier D. and Michie D. (1992). Learning to fly. In Proc. 9th Int. Machine Learning Conf. (ML'92), Aberdeen, Scotland, pp.385–393, San Mateo, CA: Morgan Kaufmann.

Savaki E. (2010). How do we understand the actions of others? By mental simulation not mirroring, Cognitive Critique, v.2, p.99-140.

Saxe R., Carey S. and Kanwisher N. (2004). Understanding other minds: Linking developmental psychology and functional neuroimaging.

Schaal S. (1999). Is imitation learning the route to humanoid robots. *Trends in Cognitive Sciences*, v.3:6, pp. 233-242.

Schaal, S., Peters, J., Nakanishi, J. and Ijspeert, A. (2005). *Learning movement primitives*, Robotics Research, Springer

Schaal S. and Sternad D. (1998). Programmable pattern generators. *In 3rd International Conference on Computational Intelligence in Neuroscience*, pp.48-51, Research Triangle Park, NC.

Schaal S., Mohajerian P. and Ijspeert A. J. (2007). Dynamic systems vs optimal control — a unifying view, *Progress in Brain Research*, v. 165, no. 1, pp. 425–445, 2007.

Schaal S, Kotosaka S, Sternad D (2000). Nonlinear dynamical systems as movement primitives. In: International conference on humanoid robotics (Humanoids00), Springer, Berlin Heidelberg New York, pp 117–124.

Schaal S. and Ijspeert A. and Billard A. (2003). Computational approaches to motor learning by imitation, The Royal Society, v.358:1431, pp.537-547.

Schoner G, Santos C (2001) Control of movement time and sequential action through attractor dynamics: a simulation study demonstrating object interception and coordination. In: Stein P, Stuart D, Selverston A (eds) Neurons, networks and motor behavior. MIT Press, Cambridge, MA.

Schutz-Bosbach S., Prinz W. (2007). Perceptual resonance: action-induced modulation of perception. *Trends Cogn. Sci,* v.11:8, pp.349–355.

Schwartz A.B., Moran D.W., Motor cortical activity during drawing movements: population representation during lemniscate tracing, J. Neurophysiol. 82 (1999) 2705–2718

Schwartz A.B. (1994). Direct cortical representation of drawing. *Science,* v.265, pp.540–542.

Scotts S.H. and Kalaska J.F. (1997). Reaching movements with similar hand paths but different arm orientations. I. Activity of individual cells in motor cortex. J. Neurophysiol, v.77, pp.826–852.

Selverston, A. I. (1993) Modeling of neuronal circuits: What have we learned? Annual Review of Neuroscience 16:531–46.

Series, P. and Latham, P.E. and Pouget, A. (2004). Tuning curve sharpening for orientation selectivity: coding efficiency and the impact of correlations, Nature neuroscience, v.7:10, pp.1129-1135.

Seung HS, Sompolinsky H (1993) Simple models for reading neuronal population codes. Proc Natl Acad Sci U S A 90: 10749–10753.

Shadmehr R, Moussavi ZM (2000) Spatial generalization from learning dynamics of reaching movements. J Neurosci 20:7807–7815.

Shadmehr R, Brashers-Krug T (1997) Functional stages in the formation of human long-term motor memory. J Neurosci 17:409–419.

Shallice, T. 1988. From Neuropsychology to Mental Structure. Cambridge Univ. Press, Cambridge.

Shidara M., Kawano K., Gomi H. & Kawato, M. Inverse-dynamics encoding of eye movement by Purkinje cells in the cerebellum. Nature 365, 50–52 (1993).

Shikata E., Tanaka Y., Nakamura H., Taira M. and Sakata, H. (1996). Selectivity of parietal visual neurons in 3D orientation of surface of stereoscopic stimuli. *Neuroreport,* v.7, pp.2389–2394.

Shultz W. (1998). Predictive reward signal of dopamine neurons. *Journal of Neurophysiology*, v.80, pp.1–27.

Shima K, Mushiake H, Saito N and Tanji, J. (1996). Role for cells in the presupplementary motor area in updating motor plans. *Proc. Natl. Acad Sci*, USA v.93, pp.8694–8698.

Shimazu H., Maier M.A., Cerri G., Kirkwood P.A., Lemon R.N. (2004). Macaque ventral premotor cortex exerts powerful facilitation of motor cortex outputs to upper limb motoneurons. *J Neurosci,* v.24, pp.1200 –1211.

Sirigu A, Cohen L, Duhamel J-R, Pi lon B, Dubois B, Agid Y, Pierrot-Deseiligny C (1995). Congruent unilateral impairments for real and imagined hand movements. *Neuroreport*, v.6, pp.997-1001.

Sirigu, A. et al. (1996). The mental representation of hand movements after parietal cortex damage. *Science*, v.273, pp.1564–1568.

Smeets J.B.J. and Denier van der Gon J.J. (1994). An unsupervised neural network for the development of reflex co-ordination. *Biol Cybern*, v.70, pp.417–442.

Song, S., Miller K.D., et al. (2000). Competitive Hebbian learning through spike-timing-dependent synaptic plasticity. *Nature Neuroscience,* v.3:9, pp.919-926.

Sokoloff, L., M. Reivich, et al. (1977). The C-deoxyglucose method for the measurement of local cerebral glucose utilization, Neurochem, v.28, pp.897-916.

Stamos V., Savaki E., and Raos, V. (2010). The spinal substrate of the suppression of action during action observation, The Journal of Neuroscience, v.30:35, pp.11605-11611.

Sternad D. and Schaal D. (1999). Segmentation of endpoint trajectories does not imply segmented control. *Exp. Brain Res,* v.124, pp.118–136.

Stein, R. B. (1967). The frequency of nerve action potential generated by applied currents. Proc. R. Soc.London, B, 167, 64-86

Stevens, C.F. and Zador, A.M. (1998). Input synchrony and the irregular firing of cortical neurons. *Nature Neuroscience*, v.1:3 pp. 210-217.

Strafella A.P., Paus T. (2000). Modulation of cortical excitability during action observation: a transcranial magnetic stimulation study. *NeuroReport,* v.11, pp.2289 –2292.

Stuart, G. J. and B. Sakmann. (1994). Active propagation of somatic action potentials into neocortical pyramidal cell dendrites, *Nature,* v.367:6458, pp.69-72.

Subiaul, F., Cantlon, J.F., Holloway, R.L. and Terrace, H.S. (2004). Cognitive imitation in rhesus macaques, American Association for the Advancement of Science, v.305:5582.

Sundara M., Namasivayam A.K., Chen R. (2001). Observation-execution matching system for speech: a magnetic stimulation study. *NeuroReport,* v.12, pp.1341–1344.

Sutton R.S., & Barto A.G. (1981). Toward a modern theory of adaptive networks: expectation and prediction. Psychol. Rev., 88, 135–170.

Sutton R.S., Singh S., Precup D. and Ravindran B. (1999). Improved switching among temporally abstract actions. *In Advances in neural information processing systems*, v.11, Cambridge, MA: MIT Press.

Synofzik M., Vosgerau G. and Newen A. (2008). I move, therefore I am: A new theoretical framework to investigate agency and ownership. *Consciousness & Cognition*, v.17, pp.411–424.

Tamura H. and Tanaka K. (2001). Visual response properties of cells in the ventral and dorsal parts of the macaque inferotemporal cortex, *Cereb. Cortex*, v.11, pp.384–399.

Tani J., Ito M. and Sugita Y. (2004). Self-organization of distributed represented multiple behavior schemata in a mirror system: Reviews of robot experiments using RNNPB. *Neural Networks*, v.17:8–9, pp.1273–1289.

Tani, J. (1998) An interpretation of the 'self' from the dynamical systems perspective: a constructivist approach. J. Conscious. Stud. 5, 516–542

Tanji J. (1975). Activity of neurons in cortical area 3a during maintenance of steady postures by the monkey. *Brain Res,* v.88, pp.549–553.

Tanji J, Taniguchi K. and Saga T. (1980). Supplementary motor area: Neuronal response to motor instructions. *J Neurophysiol*, v.43, pp.60–68.

Ter-Pogossian, M., M. Phelps, et al. (1975). A positron-emission transaxial tomograph for nuclear imaging (PETT). *Radiology,* v.114:1, pp.89-98.

Thach W. (1978). Correlation of neural discharge with pattern and force of muscular activity, joint position, and direction of intended next movement in motor cortex and cerebellum. *Journal of Neurophysiology*, v.41:3, pp.654-676.

Thelen E., Smith L. et al. (1994). A dynamic systems approach to the development of cognition and action.

Thoroughman K. A. and Shadmehr R. (2000). Learning of action through combination of motor primitives. *Nature*, v.407, pp.742-747.

Thorpe S., Delorme A. and Van Rullen R. (2001). Spike-based strategies for rapid processing, *Neural Netw*., v.14, pp.715–725.

Thorpe W.H. (1963). In Learning and instinct in animals. Cambridge, MA: Harvard University Press.

Thorpe S., Fize D. and Marlot C. (1996). Speed of processing in the human visual system, v.381:6582, pp. 520-522.

Tin C. and Poon C.S. (2005). Internal models in sensorimotor integration: perspectives from adaptive control theory, *Journal of neural engineering*, v.2, pp.s147, IOP Publishing.

Todorov E. (2003). On the role of primary motor cortex in arm movement control. *In M. L. Latash & M. Levin (Eds.), Progress in motor control III (pp. 125–166)*. Champaign, IL: Human Kinetics.

Todorov E. (2000). Direct control of muscle activation in voluntary arm movements: A model. *Nature Neuroscience*, v.3:4, pp.391–398.

Tomasello M., Savage-Rumbaugh S. and Kruger A.C. (1993). Imitative learning of actions on objects by children, chimpanzees, and enculturated chimpanzees. *Child Dev*,. v.64, pp.1688–1705.

Tomasello M. and Call J. (1997). *Primate cognition*. Oxford, UK: Oxford University Press.

Toussaint M, Goerick C. (2007). Probabilistic inference for structured planning in robotics. *In Proceedings of the IEEE/RSJ 2007 International Conference on Intelligent Robots and Systems (IROS)*, San Diego, CA, USA.

Touwen, B.C.L. (1998). The brain and development of function. *Dev. Rev*, v.18:4, pp.504-526, Elsevier.

Tsakiris M. (2010). My body in the brain: a neurocognitive model of body-ownership.

Tsakiris M., Schütz-Bosbach S. and Gallagher S. (2007). On agency and body ownership: Phenomenological and neurocognitive reflections. *Consciousness & Cognition*, v.16, pp.645–660.

Tsumoto T. and Suda K. (1979). Cross-depression: an electrophysiological manifestation of binocular competition in developing visual cortex. *Brain Research*, v.168, pp.190–194.

Turella L., Erb M., Grodd W., Castiello U. (2009). Visual features of an observed agent do not modulate human activity during action observation. *NeuroImage,* v.46, pp.844–853.

Udin, S. B., Fawcett, J. W. (1988). Formation of topographic maps. *Annu. Rev. Neurosci*. v.11: pp.289-327.

Umilta M.A., Kohler E., Gallese V., Fogassi L., Fadiga L., Keysers C. et al. (2001). I know what you are doing: A neurophysiological study. *Neuron*, v.31:1, pp.155–165.

Ungerleider L. G. and Haxby J. V. (1994). “What” and “where” in the human brain, *Curr. Opin. Neurobiol*., v.4, pp.157–165.

Ungerleider L.G. and Mishnkin, M. (2000). Two cortical visual systems.

Uno Y., Kawato M. and Suzuki R. (1989). Formation and control of optimal trajectory in human multijoint arm movement. Minimum torque-change model. *Biol Cybern,* v.61, pp.89–101.

Van Essen, D.C., Anderson, C.H. and Felleman, D.J. (1992). Information processing in the primate visual system: An integrated systems perspective. Science, v.255, pp.419-423.

van Heijst J.J., and Vos J.E. (1997). Self-organizing effects of spontaneous neural activity on the development of spinal locomotor circuits in vertebrates. Biol. Cybernet, v.77:3, PP.185--195.

VanLehn, K. (1989). Problem solving and cognitive skill acquisition. In M. Posner (Ed.), Foundations of cognitive science, pp. 527-580, Cambridge, MA: MIT Press.

Vinter A. and Perruchet P. (2002). Implicit motor learning through observational training in adults and children. *Mem Cognit,* v.30, pp.256 –261.

Vinje W.E., Gallant J.L. (2000). Sparse coding and decorrelation in primary visual cortex during natural vision. Science, v.287, pp.1273-1276.

Visalberghi E. and Fragaszy D.M. (1990). Do monkeys ape? *In ‘Language’ and intelligence in monkeys and apes (eds S. T. Parker & K. R. Gibson)*, pp.247–273, Cambridge, MA: Cambridge University Press.

Viviani P. and Terzuolo C.A. (1973). Modeling of a simple motor task in man: intentional arrest of an ongoing movement. *Kybernetik,* v.14:1, pp.35–62.

Viviani P. (2002). Motor competence in the perception of dynamic events: A tutorial. *Common mechanisms in perception and action: attention and performance*, v.19, pp.406-442.

Viviani P. and Stucchi N. (1989). The effect of movement velocity on form perception: geometric illusions in dynamic display Percept. *Psychophys*, v.46, pp.266–274.

von Bonin G. and Bailey P. (1947). The neocortex of Macaca Mulatta. Urbana: University of Illinois Press.

Vogt S. (1995). On relations between perceiving, imagining and performing in the learning of cyclical movement sequences. Br J Psychol, v.86, pp.191–216.

Vos J.E. and Scheepstra K.A. (1993). Computer-simulated neural networks: an appropriate model for motor development? *Early Hum Dev,* v.34, pp.101– 112.

Vos J.E., van Heijst J.J. and Greuters S. (1997). Programmed cell death during early development of the nervous system, modelled by pruning in a neural network. *In: Silva FL, Pr´ıncipe JC, Almeida L (eds) Spatiotemporal models in biological and artificial systems,* pp 192–199.

Wannier T.M.W., Maier M.A. and Hepp-Reymond M.C. (1989). Responses of motor cortex neurons to visual stimulation in the alert monkey. *Neu-rosci. Lett*, v.98, pp.63–68.

Webb B. (1991). Do computer simulations really cognize? *Journal of Experimental and Theoretical Artificial Intelligence,* v.3, pp.247–254.

Webb, B. (2000). What does robotics offer animal behaviour? Animal Behaviour, v.60:5, pp.545-558, Elsevier.

Weinrich M, Wise SP (1982) The premotor cortex of the monkey. J Neurosci 2:1329–45

Whiten A. et al. (1996). Imitative learning of artificial fruit processing in children (Homo sapiens) and chimpanzees (Pan troglodytes). *Journal of Comparative Psychology*, v.110:1, pp.3-14.

Whiten A. and Ham R. (1992). On the nature and evolution of imitation in the animal kingdom: reappraisal of a century of research. In Advances in the Study of Behavior, ed. PBJ Slater, JS Rosenblatt, C Beer, M Milinski, pp. 239–83, San Diego: Academic.

Wicker B. et al. (2003). Both of us disgusted in my insyla: The common neural basis of seeing and feeling disgust. *Neuron*, v.40, pp.655-664.

Widrow B. and Smith F.W. (1964). Pattern recognizing control systems. *In 1963 Comput. Inform. Sci. (COINS) Symp. Proc*, pp.288–317. Spartan.

Willner B.E., Miranker W.L., Lu C-P. (1993). Neural oganization of the locomotive oscillator. *Biol Cybern,* v.68, pp.307–320.

Wilson M. (2001). Perceiving imitable stimuli: Consequences of isomorphism between input and output. *Psychological Bulletin*, v.127:4, pp.543-553.

Wilson, C. W. (1998). Basal ganglia. In G. M. Shepherd (Ed.), *The synaptic organization of the brain*, v.3. (pp. 329–375). New York: Oxford University Press.

Wimmer H. and Perner J. (1983). Beliefs about beliefs: representation and constraining function of wrong beliefs in young children's understanding of deception *Cognition,* v.13, pp.103–128.

Wise S. and Tanji J. (1981). Neuronal responses in sensorimotor cortex to ramp displacements and maintained positions imposed on hindlimb of the unanesthetized monkey. *J Neurophysiol,* v.45, pp.482–500.

Wolpaw J. R., & Carp J. S. (1993). Adaptive plasticity in spinal cord. Adv. Neurol., 59, 163–174.

Wolpert D.M. and Kawato M. (1998). Multiple paired forward and inverse models for motor control. *Neural Networks*, v.11:7–8, pp.1317–1329.

Wolpert, D.M. (1997). Computational approaches to motor control, Trends in cognitive sciences, 1(6):209-216.

Wolpert D.M., Doya K. and Kawato M. (2003). A unifying computational framework for motor control and social interaction. *Philosophical Transactions of the Royal Society of London. Series B, Biological Sciences*, v.358:1431, pp.593–602.

Wolpert D.M., Miall R.C. and Kawato M. (1998). Internal models in the cerebellum. *Trends Cogn. Sci*, v.2, pp.338–347.

Wolpert D.M. and Kawato M. (1998). Multiple paired forward and inverse models for motor control. *Neural Networks*, v.11, pp.1317–1329.

Wolpert D.M. and Ghahramani Z. (2000). Computational principles of movement neuroscience. *Nat Neurosci [Suppl]*, v.3, pp.1212–1217.

Wolpert D.M., Ghahramani Z. and Jordan M.I. (1995). An internal model for sensorimotor integration. *Science*, v.269, pp.1880–1882.

Woosley C.N., Erickson T.C. and Gilson W.E. (1979). Localization in somatic sensory and motor areas of human cerebral cortex as determined by direct recordings of evoked potentials and electrical stimulation. *Journal of Neurosurgery*, v.51, pp.476-506.

Woosley C.N., Settlage P.H., Meyer D.R., Spencer W., Hamuy P. and Travis A.M. (1950). Patterns of localization in the precentral and “supplementary” motor areas and their relation to the concept of a premotor area. *Res. Publ. Assoc. Res. Nerv. Ment. Dis*, v.30, pp.238- 264.

Wright J.J., Robinson P.A., Rennie C.J., Gordon E., Bourke P.D., Chapman C.L., Hawthorn N., Lees G.J. and Alexander D. (2001). Toward an integrated continuum model of cerebral dynamics: The cerebral rhythms. Synchronous oscillation and conical stability, *Biosystems*, v.63, pp.71-88.

Yao, X. Evolving artificial neural networks, Proc. of the IEEE, v.87:9, pp.1423-1447. (1999).

Yue G. and Cole K.J. (1992). Strength increases from the motor program: comparison of training with maximal voluntary and imagined muscle contractions, *J. Neurophysiol*, v.67 pp.1114-1123.

Zagacki K.S., Edwards R. and Honeycutt J.M. (1992).The role of mental imagery and emotion in imagined interaction, *Communication Quarterly*, v.40:1, pp.56-68, Taylor and Francis.

Zeki S., Watson J. et al. (1991). A direct demonstration of functional specialization in human visual cortex, Journal of Neuroscience, v.11:3, pp.641-649.

Zhang, X., J. Zhang, et al. (1999). "A parallel distributed processing model of stimulus-stimulus and stimulus response compatibility." Cognitive Psychology 38: 386-432.

Zhang K., Ginzburg I., McNaughton B.L., Sejnowski T.J., Interpreting neuronal population activity by reconstruction: unified framework with application to hippocampal place cells, J. Neurophysiol. 9 (1998) 1017–1044.

Zhou Y-D, Fuster J (1996) Mnemonic neuronal activity in somatosensory cortex. Proc Natl Acad Sci USA 93:10533–10537

Zipser D. and Andersen R.A. (1988). A back-propagation programmed network that simulates response properties of a subset of posterior parietal neurons, *Nature*, v.331, pp. 679-684.

Zipster D., (1992). Identification models of the nervous system, *Neuroscience*, v.47:4, pp.853-862, Elsevier.

Zucker, R.S. (1989). Short-term synaptic plasticity, Annual Reviews, v.12:1, pp.13-31.